

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

Кафедра системного програмування і спеціалізованих комп'ютерних систем

До захисту допущено

Завідувач кафедри

\_\_\_\_\_  
(підпис)                      Віталій РОМАНКЕВИЧ  
(ініціали, прізвище)

“ \_\_\_\_ ” червня 2020 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Системне програмування»  
спеціальності                      123 «Комп'ютерна інженерія»**

на тему: «Система організації та контролю спортивних і навчальних занять»

Виконав: студент IV курсу, групи KB-63

Філатов Олексій Олександрович

\_\_\_\_\_  
(підпис)

Керівник ст..викладач каф. СПСКС Наливайчук М. В.

\_\_\_\_\_  
(підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М.

\_\_\_\_\_  
(підпис)

Рецензент \_\_\_\_\_

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Віталій РОМАНКЕВИЧ  
(підпис) (ініціали, прізвище)

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на дипломний проєкт студента**

Філатова Олексія Олександровича

1. Тема проєкту «Система організації та контролю спортивних і навчальних занять», керівник проєкту Наливайчук Микола Васильович, ст.викладач каф. СПСКС, затверджені наказом по університету від 25 травня 2020 р. № 1181-С

2. Термін подання студентом проєкту \_\_\_\_\_

3. Вихідні дані до проєкту: див. Технічне завдання

4. Зміст пояснювальної записки:

- аналіз існуючих рішень та обґрунтування теми дипломного проєкту;
- особливості розробки додатку на Android-платформі;
- реалізація функції збереження інформації Android-додатку за допомогою СУБД SQLite;
- особливості реалізації Android-додатку для контролю навчального та спортивного процесу.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо):

- алгоритм запису нового найменування у базу даних (схема алгоритму);

- схема зв'язку таблиць, що формують базу даних (схема структурна);
- алгоритм виводу поточної події (схема алгоритму);
- схема взаємозв'язків модулю Train (схема структурна);
- презентація за темою роботи.

#### 6. Консультанти розділів проєкту\*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М. доц. каф. СПСКС, к.т.н.		

7. Дата видачі завдання \_\_\_\_\_

#### Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення літератури за тематикою проєкту	02.03.2020	
2	Розроблення технічного завдання	12.03.2020	
3	Аналіз існуючих рішень	20.03.2020	
4	Підготовка матеріалів першого розділу дипломного проєкту	21.03.2020	
5	Вивчення додаткової літератури	30.03.2020	
6	Розробка додатку	15.04.2020	
7	Відлагодження та тестування додатку	30.04.2020	
8	Підготовка матеріалів другого розділу дипломного проєкту	05.05.2020	
9	Підготовка матеріалів третього розділу дипломного проєкту	10.05.2020	
10	Підготовка матеріалів четвертого розділу дипломного проєкту	15.05.2020	
11	Підготовка матеріалів графічної частини проєкту	17.05.2020	
12	Оформлення технічної документації проєкту	18.05.2020	
13	Попередній захист проєкту	20.05.2020	

Студент

\_\_\_\_\_ (підпис)

Олексій Філатов

Керівник проєкту

\_\_\_\_\_ (підпис)

Микола Наливайчук

\*Консультантом не може бути зазначено керівника дипломного проєкту.

## АНОТАЦІЯ

Об'єктом розробки є система для організації та контролю спортивних та навчальних занять.

Дана система виготовлена у вигляді додатку на операційній платформі Android. Додаток дозволяє користувачеві просто і швидко занотовувати свої персональні тренування, свої вправи та списки занять, та синхронізувати ці дані з календарем. Для зберігання даних на пристрої додаток використовує базу даних SQLite. В процесі розробки додатку:

- був проведений аналіз існуючих додатків подібного типу, та розбір переваг і недоліків кожного з типових;
  - сформовано основні вимоги до розроблюваного додатку;
  - автор ознайомився з основними можливостями середовища розробки;
  - розроблено графічний інтерфейс додатку;
  - розроблено методи для оперування базою даних;
  - розроблена програмна частина, що зв'язує графічний інтерфейс та базу даних.

Цей додаток поєднує у собі риси спортивного та навчального помічника, і буде корисним для активного студента.

Ключові слова:

Android studio, Android додаток, SQLite, спортивний помічник, навчальний помічник, календар.

## ANNOTATION

Sport and study activity control system is the main aim of this development. This system is developed as an Android application. Developed application allows user to save his own trainings, exercise and study lists and also provides a possibility to connect this list with some day in the calendar. This application use SQLite database as a data storage. During the work:

- author made an analysis based on existing types of application, and look through all their advantages and disadvantages;
- author figured out with Android Studio possibilities;
- graphic interface was developed;
- methods for database were developed;
- author made a part, which connects database and interface.

This application has all advantages from sport and study assistant combination.

Key words:

Android studio, Android application, SQLite, sport assistant, study assistant, calendar.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
			<u>Документація загальна</u>			
			<u>Новорозроблена</u>			
	A4	ІАЛЦ.045490.002 ТЗ	Система організації та контролю спортивних і навчальних занять.	3		
			Технічне завдання.			
	A4	ІАЛЦ.045490.003 ТП	Система організації та контролю спортивних і навчальних занять.	1		
			Відомість технічного проєкту.			
	A4	ІАЛЦ.045490.004 ПЗ	Система організації та контролю спортивних і навчальних занять.	50		
			Пояснювальна записка.			
	A4	ІАЛЦ.045490.005 Д1	Запис нового найменування у базу	1		
			Схема алгоритму			
	A4	ІАЛЦ.045490.006 Д2	Зв'язок таблиць, що формують базу даних.	1		
			Схема структурна			
Змін.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.045490.001 ОА	
Розробив	Філатов О.О.				Система організації та контролю спортивних і навчальних занять. Опис альбому	
Перевірив	Наливайчук М.В.					
Н. контроль	Клятченко Я.М.					
Зав. каф.	Романкевич В.О.				Літ.      Аркуш      Аркушів	
					1	2
					КПІ ім. Ігоря Сікорського, ФПМ KB-63	

[illegible]

## ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ. ....	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ. ....	2
3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ. ....	2
4. ДЖЕРЕЛА РОЗРОБКИ. ....	2
5. ТЕХНІЧНІ ВИМОГИ. ....	3
5.1. Вимоги до додатку, що розробляється. ....	3
5.2. Вимоги до програмного забезпечення користувача ....	3
6. ЕТАПИ РОЗРОБКИ. ....	3

					ІАЛЦ.045490.002 ТЗ								
Зм.	Арк	№ докум	Підпис	Дата									
Розроб.		Філатов О.О.			Система організації та контролю спортивних і навчальних занять. Технічне завдання				Літ.		Арк.	Аркушів	
Перев.		Наливайчук М.В.										1	3
									КП ім. Ігоря Сікорського, ФПМ КВ-63				
Н. контр.		Клятченко Я.М.											
Затв.		Романкевич. В.О.											



## 1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Система організації та контролю спортивних і навчальних занять».

Галузь застосування: організація та контроль навчальних та спортивних занять.

## 2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

## 3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є розробка та створення додатку для контролю та організації спортивних і навчальних занять.

## 4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації даного проєкту є науково-технічна та технічна література, технічна документація, учбові посібники та електронні статті у мережі інтернет.

					ІАЛЦ.045490.002 ТЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підп.	Дата		

## 5. ТЕХНІЧНІ ВИМОГИ

### 5.1. Вимоги до додатку, що розробляється:

- наявність функції запису тренувань та предметів;
- коректне відображення існуючого списку тренувань та предметів;
- можливість редагувати списки;
- можливість перегляду списку існуючих вправ та його редагування;
- можливість перегляду заданих вправ у тренуванні;
- наявність зручного користувацького інтерфейсу;
- мінімальний об'єм зайнятої пам'яті.

### 5.2. Вимоги до програмного забезпечення користувача:

- операційна система Android версії 6.0 і вище.

## 6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення літератури за тематикою проєкту	02.03.2020	
2	Розроблення технічного завдання	12.03.2020	
3	Аналіз існуючих рішень	20.03.2020	
4	Підготовка матеріалів першого розділу дипломного проєкту	21.03.2020	
5	Вивчення додаткової літератури	30.03.2020	
6	Розробка додатку	15.04.2020	
7	Відлагодження та тестування додатку	30.04.2020	
8	Підготовка матеріалів другого розділу дипломного проєкту	05.05.2020	
9	Підготовка матеріалів третього розділу дипломного проєкту	10.05.2020	
10	Підготовка матеріалів четвертого розділу дипломного проєкту	15.05.2020	
11	Підготовка матеріалів графічної частини проєкту	17.05.2020	
12	Оформлення технічної документації проєкту	18.05.2020	
13	Попередній захист проєкту	20.05.2020	

Зм.	Арк.	№ докум.	Підп.	Дата

ІАЛЦ.045490.002 ТЗ

Арк.  
3



## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ТЕРМІНІВ, ТЕРМІНІВ, СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ.....	6
1.1. Загальний опис проблеми.....	6
1.2. Аналіз існуючих додатків для контролю спортивних і навчальних занять.....	7
1.3. Обґрунтування теми диплому.....	12
2. ОСОБЛИВОСТІ РОЗРОБКИ ДОДАТКУ НА ANDROID-ПЛАТФОРМИ.....	13
2.1. Опис мобільної платформи Android.....	13
2.2. Опис середовища розробки Android Studio.....	15
2.3. Особливості середовища розробки.....	17
2.3.1 Система автоматичного збирання Gradle.....	17
2.3.2 Комплект розробки програмного забезпечення Android SDK.....	18
2.3.3. Менеджер віртуальних пристроїв AVD Manager.....	19
2.3.4. Помічник налагоджувача Logcat.....	21
2.4. Особливості розробки користувацького інтерфейсу програми.....	22
2.4.1. Формування програмної частини за допомогою компоненту Activity.....	22
2.4.2. Формування основної графічної частини за допомогою XML-файлу.....	25
3. РЕАЛІЗАЦІЯ ФУНКЦІЇ ЗБЕРЕЖЕННЯ ІНФОРМАЦІЇ ANDROID-ДОДАТКУ ЗА ДОПОМОГОЮ СУБД SQLITE.....	28
3.1 Опис системи управління базами даних SQLite.....	28

					ІАЛЦ.045490.004 ПЗ			
Зм.	Арк	№ докум	Підпис	Дата	Система організації та контролю спортивних і навчальних занять Пояснювальна записка			
Розроб.		Філатов О.О.						
Перев.		Наливайчук М.В.						
Н. контр.		Клятченко Я.М.						
Затв.		Романкевич. В.О.						
					Літ.		Арк.	Аркушів
							1	50
					КПІ ім. Ігоря Сікорського, ФПМ KB-63			

3.2 Реалізація бази даних на Android за допомогою SQLite.....	28
4. ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ANDROID-ДОДАТКУ ДЛЯ КОНТРОЛЮ НАВЧАЛЬНОГО ТА СПОРТИВНОГО ПРОЦЕСУ.....	31
4.1 Реалізація головного меню додатку.....	31
4.2 Реалізація системи контролю спортивного процесу.....	32
4.2.1 Реалізація меню вибору тренувань та вправ.....	32
4.2.2 Реалізація відображення та редагування списку тренувань.....	33
4.2.3 Реалізація відображення та редагування списку вправ.....	41
4.3 Реалізація системи контролю навчального процесу.....	42
4.4 Реалізація системи моніторингу за тренуваннями та навчальною діяльністю.....	44
4.5 Реалізація та конфігурація бази даних тренувань за допомогою класу TrainHelper.....	45
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	49
ДОДАТКИ	

Додаток 1. Копії графічних матеріалів.

- ІАЛЦ.045490.005 Д1. Запис нового найменування у базу даних.  
Схема алгоритму
- ІАЛЦ.045490.006 Д2. Зв'язок таблиць, що формують базу даних.  
Схема структурна
- ІАЛЦ.045490.007 Д3. Вивід поточної події. Схема алгоритму
- ІАЛЦ.045490.008 Д4. Взаємозв'язки модулю Train. Схема структурна

Додаток 2. Лістинг програми.

Додаток 3. Презентація.

## ПЕРЕЛІК УМОВНИХ ТЕРМІНІВ, ТЕРМІНІВ, СКОРОЧЕНЬ

1. Лог – спеціалізована програмна інформація ;
2. Рефакторинг – перетворення програмного коду;
3. Розумний годинник – наручний комп’ютеризований годинник з розширеним функціоналом;
4. СУБД – система управління базами даних;
5. Утиліта – сервісна програма;
6. ADB(Android Debug Bridge) – інструмент для керування пристроєм на платформі Android;
7. Android SDK - це комплект розробника, завдяки якому створюються нові додатки для операційної системи Android;
8. Apache Ant – утиліта для автоматизації процесу компіляції програми;
9. APK(Android Package) – формат архіву, що використовується операційною системою Android для встановлення і розповсюдження програм;
10. GitHub – сервіс для спільної розробки;
11. GO – мова програмування, розроблена компанією Google;
12. iOS – мобільна операційна система;
13. Mac OS – операційна система для комп’ютера;
14. Play Market – Android магазин додатків, ігор, книг, музики та фільмів;
15. XML(Extensible Markup Language) – розширювана мова розмітки.

					ІАЛЦ. 045490.004 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підп.	Дата		

## ВСТУП

На сьогоднішній день, активне проведення свого вільного часу вельми популярне. Учні шкіл, студенти, дорослі люди і, навіть, пенсіонери намагаються проводити вільний час з користю для себе і свого здоров'я. Здоровий образ життя все частіше і частіше займає провідне місце у розкладі кожної людини. Це можуть бути заняття важкими фізичними навантаженнями, різними видами спорту або ранкова пробіжка. Для задоволення потреб таких людей існує дуже багато тренажерних залів різного розміру, статусу та напрямлення. Професійні багатоярусні зали, де займаються різні типи людей, і, розкидані по спальним районам, маленькі тренажерні зали, які розраховані на невелику кількість людей, і мають цілі династії постійних відвідувачів.

Проте з розвитком рівня технологій, споживач став потребувати більш тонкого, простого, швидкого доступного способу записувати свої плани тренування та спортивні досягнення. З появою телефонів на різних операційних системах, почали з'являтися і додатки, що були своєрідними менеджерами часу. Проте вони були недосконалі, займали багато пам'яті телефону та мали безліч недоліків. Але згодом всі пристрої на операційних системах зазнали уніфікації, і основний ринок подібних телефонів розділився на два великих табори: пристрої на операційній платформі Android і пристрої на операційній платформі iOS. Завдяки цьому розділенню, досить швидко на просторах інтернету стали з'являтися додатки, які могли задовольнити потребу кожного користувача.

Навчальний процес також потребував технології для запису кількості та найменування предметів. Звичайно, в основному це стосується студентів університетів, коледжів, різних інших закладів вищої освіти. Бо учні шкіл все ще мають архаїчний щоденник, хоча в деяких школах та ліцеях щоденник вже існує у форматі глобальної мережі. Завдяки цьому учні мають можливість слідкувати за порядком уроків та їх кількістю, бачити зміни у домашньому завданні. Студенти вищих учбових закладів мають сайти, завдяки яким у них є можливість

					ІАЛЦ. 045490.004 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підп.	Дата		

дізнатися розклад пар. Проте не завжди є можливість з'єднатися з глобальною мережею інтернет, а отже і користі існуючі сайти принести не здатні. Відповідно, і програмне забезпечення не повинно залежати від рівня зв'язку з глобальною мережею інтернет.

Додаток, створений у процесі роботи над дипломним проєктом, дозволяє користувачам швидко і зручно записувати у пам'ять телефону плани тренувань та розклад навчальних занять. На думку автора, додаток, що буде створений у процесі роботи над проєктом може стати помічником для основної цільової аудиторії цього продукту – студентів. Додаток не використовує глобальну мережу, і займає мінімальну кількість пам'яті пристрою. Це дозволяє додатку оперувати даними без затримок.

					ІАЛЦ. 045490.004 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підп.	Дата		



# 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

## 1.1. Загальний опис проблеми.

На даний момент існує велика кількість програм для управління своїм часом та розпорядком дня: від простих блокнотів, записників і календарів, до складних систем з обліковими записами і монетизацією. На відміну від простих, складні додатки та їх розробники мають за мету комерційний прибуток, а отже і будь-які інші цілі цього продукту є вторинними, і весь продукт, в першу чергу, розробляється з метою отримати прибуток. Проте навіть таких видів додатку існує декілька.

Перший вид – це платні додатки, тобто такі, які можна придбати за допомогою спеціальних платформ для поширення програм, наприклад Play Market. Такі додатки мають різну ціну, проте перевагою є те, що після купівлі користувачу доступний увесь спектр функцій даної програми. Нерідко такі програми багатофункціональні та мають підтримку фітнес-браслетів.

Другий вид – це безкоштовні додатки, але з частиною заблокованого функціоналу, який можна розблокувати тільки придбанням платної версії. Проте на відміну від першого, в другому безкоштовний функціонал присутній, що не дає гарантії на налагоджену роботу додатка.

Третій вид – це безкоштовні додатки, ці додатки не мають платного функціоналу і знаходяться у вільному доступі. Проте такий додаток має один серйозний недолік – перенасиченість рекламою. Рекламні оголошення можуть займати до 30% екранного простору. Така ситуація впливає на зручність та інформаційну завантаженість екрану, та негативно впливає на сприйняття додатка в цілому. На думку автора, цей вид має найбільший недолік, у порівнянні з іншими.

Але на відміну від комерційних проектів, існують і некомерційні, які досить часто розроблюються командами волонтерів, або студентами, в якості

					ІАЛЦ. 045490.004 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підп.	Дата		

практики та самостійного вивчення певної проблеми. Подібні додатки є не оптимізованими та нерідко займають значну кількість пам'яті пристрою, що негативно позначається на розповсюдженні продукту. Проте такі проєкти залишаються непопулярними, і згодом втрачають підтримку розробника та користувача.

## 1.2. Аналіз існуючих додатків для контролю спортивних і навчальних занять.

Серед великого числа існуючих додатків для організації та контролю спортивних занять можна виділити найбільш типові з них: FitProSport, Nike+ Training Club, Jefit. Потрібно зазначити, що майже всі подібні додатки потребують реєстрації, що для деяких користувачів може стати проблемою.

- FitProSport(Рис 1.2.1)



Рис. 1.2.1- Интерфейс додатку FitProSport

- + Швидкість роботи цього додатку середня, затримок у роботі виявлено не було;
- + Користувацький інтерфейс простий і орієнтований на споживача;
- + Додаток має значну кількість програм тренувань, довідку по кожній вправі, можливість створення свого плану тренування, або своєї вправи.(Рис. 1.2.2);
- Перенасиченість рекламною інформацією;
- Для отримання розширеного функціоналу необхідно придбати повну версію програми;



Рис. 1.2.2 - Приклад довідки по вправі

- Nike+ Training Club
  - + Повністю безкоштовний додаток.
  - + Наявність в базі даних 150 тренувань для усіх груп м'язів;

					ІАЛЦ. 045490.004 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підп.	Дата		

- + Відсутність рекламних матеріалів;
- + Зручний дизайн (Рис. 1.2.3);
- + Зручний вибір цілі тренування;
- Додаток займає великий об'єм пам'яті пристрою;
- Швидкодія додатку на низькому рівні через велику завантаженість оперативної пам'яті пристрою.



Рис. 1.2.3 - Інтерфейс додатку Nike+ Training Club

- Jefit
  - + Велика кількість програм з тренуваннями для фітнесу та важкої атлетики;

- + Наявність довідника для вправ;
- Складний інтерфейс перевантажений рекламними оголошеннями(Рис. 1.2.4);
- Недостатня локалізація додатку;
- Необхідність реєстрації;
- Нав'язування платної версії.

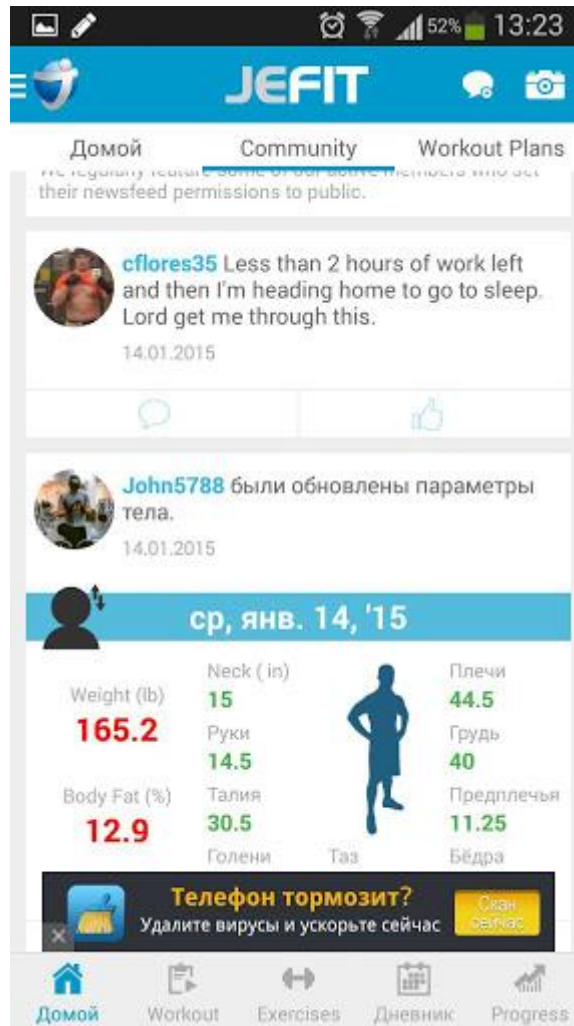


Рис. 1.2.4 - Приклад інтерфейсу додатку Jefit

На відміну від додатків спортивного напрямку, додатки для адміністрування навчального процесу досить схожі. Усі програми подібного типу є таблицями або календарем з розширеним функціоналом. Варто розглянути один з додатків даного типу. Наприклад, додаток TimeTable.

					ІАЛЦ. 045490.004 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підп.	Дата		

- + Формування розкладу з аудиторіями, викладачами і часом;
- + Можливість прикріпити фотографію, документ або голосове повідомлення;
- Низький рівень швидкодії;
- Інтерфейс програми складний та перенасичений кольоровою гаммою(Рис. 1.2.5);
- Наявність нав'язування платної версії.



Рис. 1.2.5 - Приклад інтерфейсу додатку TimeTable

На думку автора, кожна із вище зазначених програм перенасичена зайвими функціями і займає значний об'єм корисної пам'яті. Деякі додатки мають у своєму складі велику кількість рекламних оголошень, що негативно впливає на

					ІАЛЦ. 045490.004 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підп.	Дата		

зручність додатку та його рейтинг. Додатків, які мають мінімалістичний дизайн і займають малий об'єм корисної пам'яті не така велика кількість. В основному це – блокноти або нотатки, вбудовані в операційну систему пристрою. Відповідно в цих додатках відсутня можливість синхронізації даних з календарем та база даних тренувань.

### 1.3. Обґрунтування теми диплому.

Головна мета дипломного проєкту – створення додатку для контролю та організації навчальних та спортивних занять. На даний момент не існує окремого додатку для задоволення обох потреб : запису бази тренувань та необхідних предметів. Після аналізу існуючих додатків, врахувавши всі переваги та недоліки кожного додатку з схожою тематикою, автором були розроблені основні вимоги до додатку подібного типу:

- Висока швидкодія та малий об'єм використовуваної пам'яті пристрою;
- Простий користувацький інтерфейс, зручний у користуванні;
- Відсутність рекламних оголошень та нав'язування платної версії;
- Мінімалістичний дизайн з відсутністю зайвих елементів;
- Наявність списку базових вправ;
- Об'єднання функції запису плану тренувань та функції запису плану навчання.

Подібний додаток має використовуватися, як допоміжна система, орієнтована, в першу чергу, на активного студента. Мета даного проєкту – додаток на платформі Android, який об'єднає в собі риси спортивного помічника та розклад занять зі списком необхідних студенту предметів.

					ІАЛЦ. 045490.004 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підп.	Дата		

## 2. ОСОБЛИВОСТІ РОЗРОБКИ ДОДАТКУ НА ANDROID-ПЛАТФОРМИ.

### 2.1. Опис мобільної платформи Android.

Android – це операційна система, розроблена для багатьох видів пристроїв. Серед них: мобільні телефони, планшетні комп'ютери, розумні годинники, електронні книги, ігрові приставки, ноутбуки та багато других пристроїв. У 2015 році на платформі Android почали з'являтися наручні годинники, телевізори з постійними зв'язком з глобальною мережею і, навіть, автомобілі з керуванням за допомогою систем штучного інтелекту.

Операційна система заснована на модернізованому ядрі Linux і використовує власну віртуальну машину Java. Головним авторами цієї операційної системи є Енді Рубін, Нік Сірс, Річард Майнер та Кріс Уайт. Початковою метою компанії було створення розширеної операційної системи для цифрових фотокамер. Згодом автори Android усвідомили недоцільність створення окремої операційної системи для фотоапаратів, та змінили свою мету на більш глобальну – створення операційної системи зручної у керуванні з можливістю подальшого розвитку(Рис. 2.1.1).



Рис. 2.1.1 - Перший логотип операційної системи Android

Але у 2005 році через нестачу фінансування та відсутність спонсорів компанія Android.inc стала власністю компанії Google, з якою і розпочала своє довготривале співробітництво. Масовий результат цього співробітництва вперше з'явився у вересні 2008 року на основі мобільного телефону HTC Dream (Рис. 2.1.2). З 2009 року операційна система Android зазнала великої кількості змін та оновлень, які прибрали основні недоліки цієї системи та додали безліч

					ІАЛЦ. 045490.004 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підп.	Дата		



нових корисних функцій. Операційна система набула вигляд більш зрозумілої та легкої у користуванні, підвищився комфорт користувача.



Рис.2.1.2 - Перший пристрій реалізований на операційній системі Android - HTC Dream

У 2017 році дана операційна система налічувала більше 2 мільярдів користувачів, а на березень 2020 кількість доступних додатків на Android налічувала більше 2.9 мільйона одиниць [1].

Інтерфейс операційної системи Android інтуїтивно зрозумілий та використовує прості тактильні дії для маніпулювання операціями і об'єктами на екрані, та віртуальну клавіатуру, за допомогою якої доступна можливість набирати текст повідомлення. Для забезпечення коректної роботи дана операційна система використовує додаткові апаратні засоби: акселерометр, гіроскоп та оптичні датчики.

Додатки на операційну систему розроблені на основі таких мов програмування, як Kotlin, Java, і C++ та зкомпільовані за допомогою набору розробника - Android SDK у файл формату APK. Kotlin – це мова програмування, яка з'явилася з метою створення більш простого і в той же час лаконічного

					ІАЛЦ. 045490.004 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підп.	Дата		

варіанту Java. Це статично типізована, об'єктно-орієнтована мова програмування, що працює завдяки віртуальній машині Java. Розробником цієї мови стала компанія JetBrains, на основі середовища розробки якої й було побудоване середовище розробки Android Studio.

Завдяки відкритому коду та наявності майже усіх бібліотек і середовищ розробки у вільному доступі, додатки будь-якого типу і цілі здатні з'явитися на операційній системі Android. Також завдяки цьому існує велика кількість розробників додатків на даній системі, а звідси і велика кількість комерційних проектів, які можуть постійно підтримувати актуальність свого додатку, завдяки постійному оновленню баз даних Android.

До 2013 року основним середовищем розробки для операційної системи Android було Eclipse, але вже згодом з'явилося Android Studio, яке мало велику кількість переваг. Особливо зручним для розробників є середовище розробки Android Studio через наявність у ньому можливостей використовувати мови програмування зазначені вище, наявність Android SDK, наявність емулятора віртуальних пристроїв AVD та інших корисних утиліт.

## 2.2. Опис середовища розробки Android Studio

Android Studio – це офіційне середовище розробки для операційної системи Android, побудоване на основі середовища розробки IntelliJ IDEA. Розробником даного середовища є компанія JetBrains сумісно з компанією Google. Дане середовище розробки підтримує такі популярні мови програмування, як Java та C++, та більш спеціалізовані такі, як Kotlin і GO.

Android Studio доступне для встановлення на всі популярні операційні системи: Windows, macOS і Linux. Основними перевагами даного середовища розробки є:

- Система автоматичної побудови проекту Gradle;
- Велика кількість графічних макетів для коректного відображення графічного інтерфейсу майбутньої програми(Рис. 2.2.1);

					ІАЛЦ. 045490.004 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підп.	Дата		

- Підтримка розробки додатків на наручні годинники виготовлені на операційній системі Android;
  - Емулятор віртуальних пристроїв, за допомогою якого розробник може адаптувати свій додаток під різні типи пристроїв та види екранів;
  - Вбудована технологія Google Cloud Platform, що реалізує зберігання необхідних вам даних та постійну підтримку середовища;
  - Велика кількість інструментів для побудови, аналізу та компіляції проєкту;
  - Корисні підказки для розробників-початківців;
  - Наявність великої бібліотеки елементів XML;
  - Підтримка C++;
  - Можливість синхронізувати свої дії при роботі у команді за допомогою GitHub;
  - Поглиблений рефакторинг та швидкі зміни коду;
  - Розумна система редактора коду, що підказує який елемент потрібно підключити;
  - Наявність зручних інструментів для тестування;
  - Наявність логів для відстеження помилок, що виникли при компіляції, налагодженні та роботі програми;
  - Можливість внесення змін та їх відображення без перезапуску програми.
- Проте у даного середовища розробки є і недоліки:
- Затримки у обміні інформацією;
  - Затримки у компіляції та побудові файлу;
  - Некоректна робота емулятора віртуальних пристроїв, через що неможливо визначити причину несподіваного припинення роботи додатку та пристрою в цілому;
  - Високі системні вимоги даного середовища;
  - Відсутність у деяких моментах коректного підключення бібліотек та зовнішніх файлів [2].

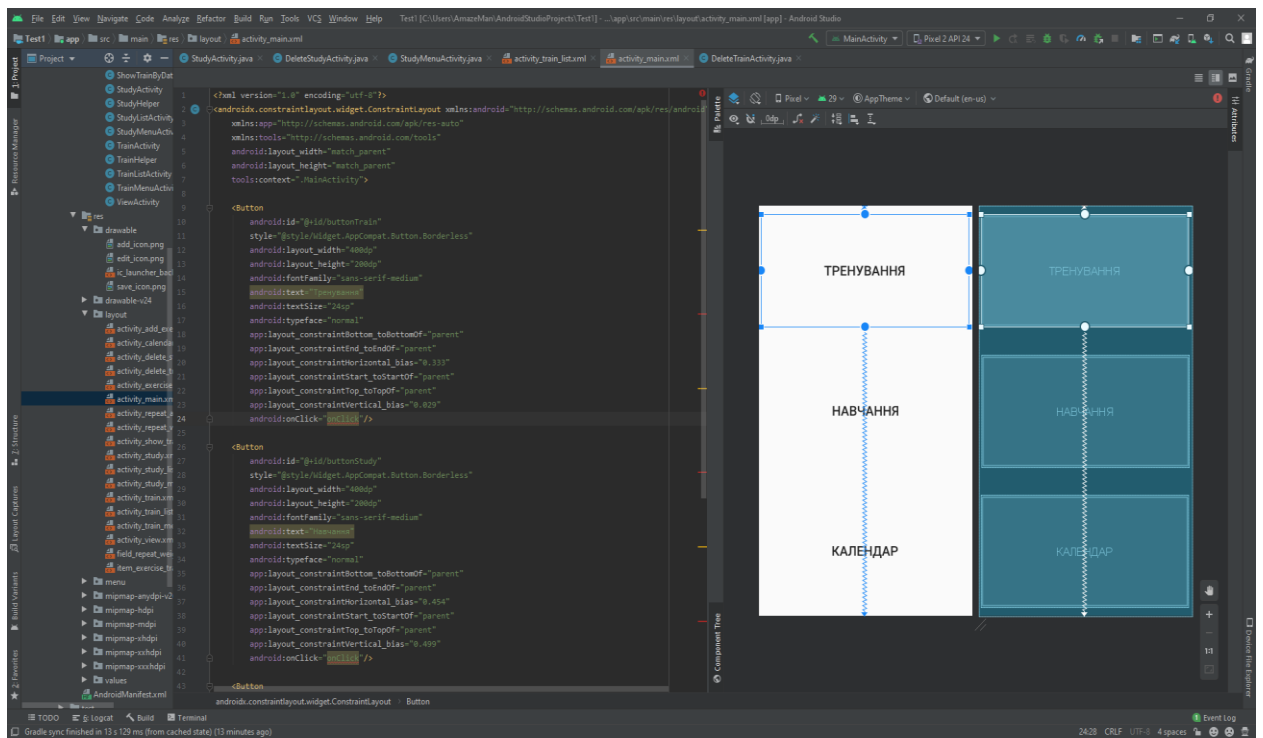


Рис. 2.2.1 - Приклад інтерфейсу середовища розробки Android Studio

Проте наявність корисних підказок, що дійсно допомагають користувачеві розібратися у інтерфейсі програми, допомогла автору швидко розібратися у даному середовищі розробки.

## 2.3. Особливості середовища розробки

### 2.3.1 Система автоматичного збирання Gradle

Gradle – система автоматичного збирання, яка побудована на концептах Apache Ant і Apache Maven, і використовує предметно-орієнтовану мову на базі мови Groovy замість традиційної мови розмітки XML. Gradle використовує орієнтований ациклічний граф для визначення порядку виконання завдань. Дана система зв'язує завдання разом згідно з залежностями для створення графу. Як тільки граф був побудований, Gradle визначає які завдання потрібно виконати у якому порядку, і приступає до виконання [3].

Діаграма показує нам два приклади графу : перший – абстрактний і другий – конкретний з зазначенням залежностей за допомогою стрілок(Рис

					ІАЛЦ. 045490.004 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підп.	Дата		

2.3.1.1). Gradle була створена для побудови та розробки проєктів, які згодом зможуть розростатися у об'ємі завдяки функції інкрементування. Дана система працює в активній віртуальній машині Java, використовує бібліотеки Ant та інші базові інструменти.

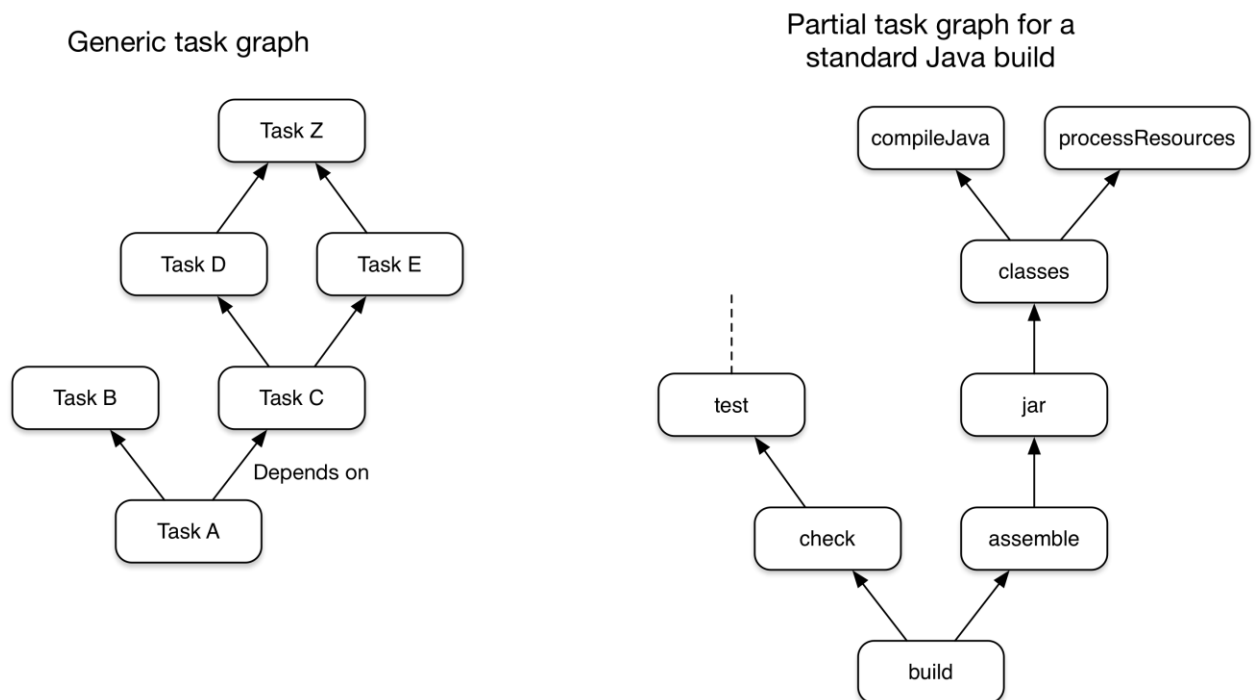


Рис 2.3.1.1 - Два приклади графів завдань, що формує Gradle

## 2.3.2 Комплект розробки програмного забезпечення Android SDK

Android SDK – це комплект розробника, завдяки якому створюються нові додатки для операційної системи Android. Даний набір включає в себе:

- Бібліотеки, необхідні для підтримки різних версій;
- Налаштовувач коду;
- Утиліта ADB(Android Debug Bridge);
- Помічник налагоджувача Logcat, який дає можливість спостерігати за плином помилок у реальному часі;
- Приклади різних програм;
- Навчальні матеріали [4].

Середовище Android SDK має можливість коректно функціонувати на комп'ютерах, що використовують операційну систему Mac OS X 10.5.8 і більш нові її версії, Windows 7 і більш нові її версії та Linux. Даний комплект також підтримує старі версії операційної системи Android, для розробників, які мають на меті підтримку споживачів зі старими пристроями.

Інструменти Android SDK компілюють існуючий код, існуючий пакет даних і ресурсів і формують файл з розширенням APK. В даному файлі знаходяться всі необхідні ресурси необхідні для коректного функціонування Android-додатку, і цей файл дає можливість встановити даний додаток на будь-який пристрій, що працює на операційній системі Android.

### 2.3.3. Менеджер віртуальних пристроїв AVD Manager

AVD Manager – допоміжна утиліта, яка дозволяє розробнику протестувати програму у режимі реального часу за допомогою емулятора різних типів пристрою(Рис. 2.3.3.1).

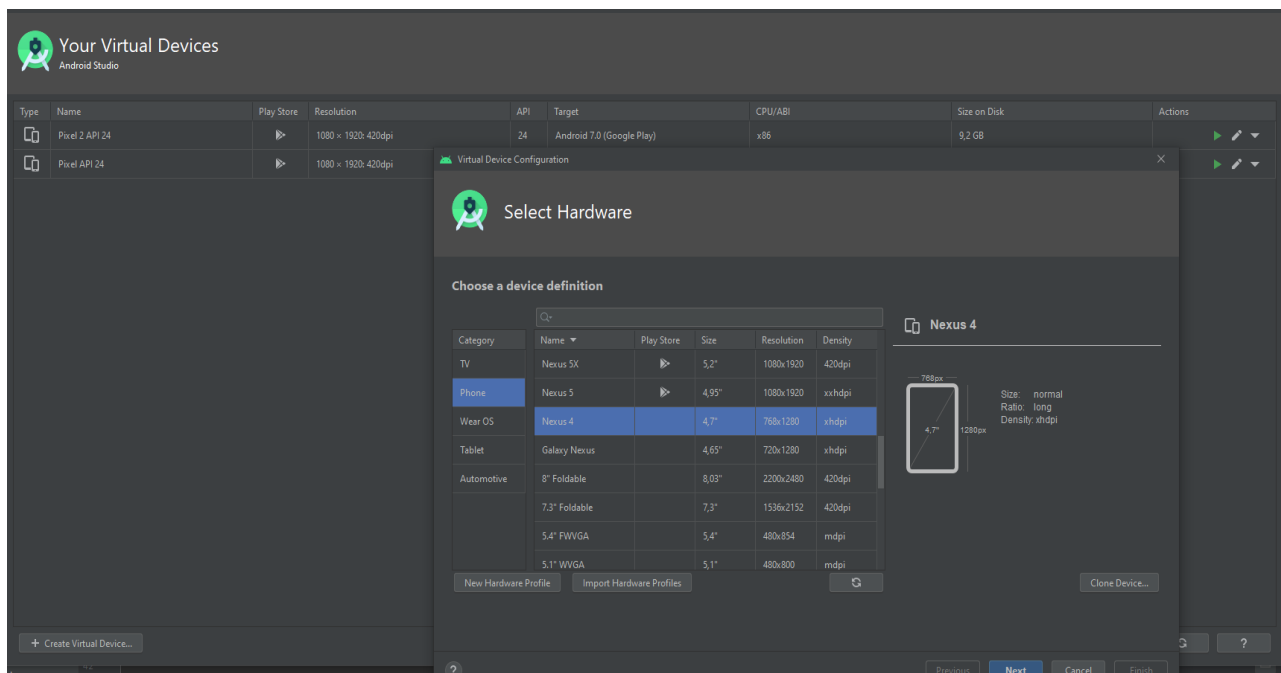


Рис. 2.3.3.1- Інтерфейс утиліти AVD Manager

Даний інструмент також існує для оновлення, налаштування та перевірки елементів Android SDK. За допомогою цієї утиліти розробник має можливість створити будь-який пристрій з будь-яким розміром екрану, технічними характеристиками та версією операційної системи Android (Рис. 2.3.3.2). При виборі ще не завантаженої версії, AVD Manager самостійно завантажить і встановить потрібну версію на віртуальний пристрій.

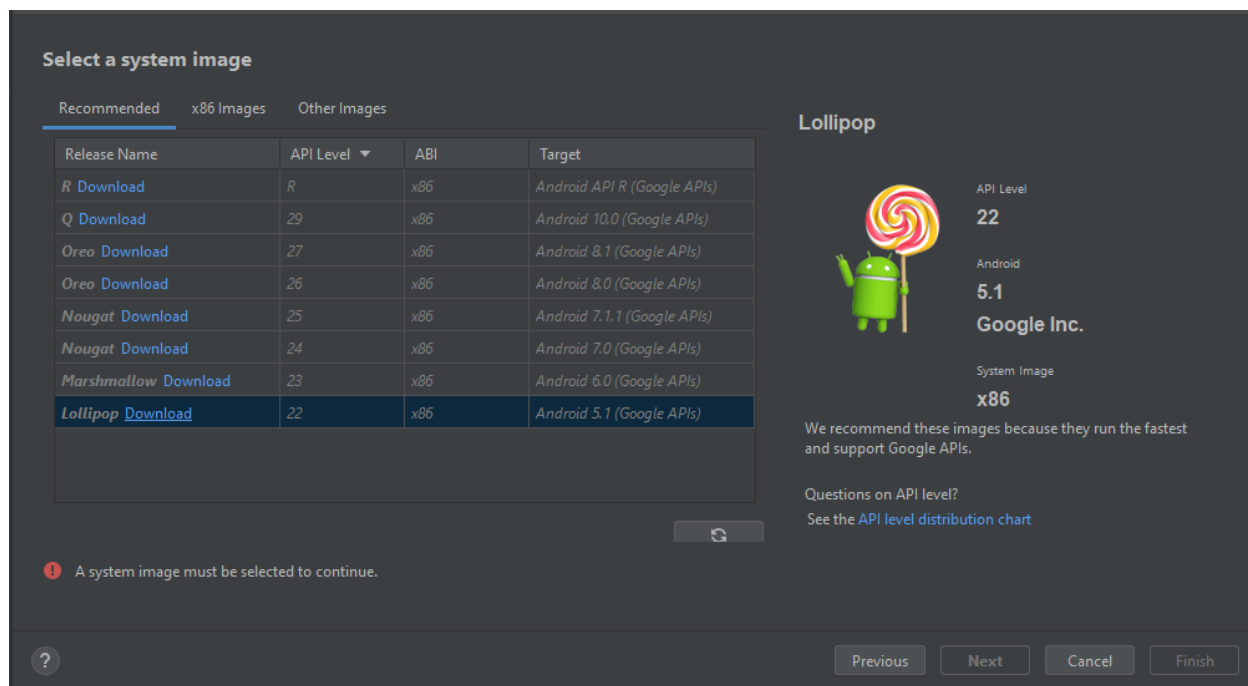


Рис. 2.3.3.2 - Можливості вибору операційної системи Android

Після усіх необхідних налаштувань і запуску емулятора, розробник отримує доступ до симуляції Android-пристрою (Рис. 2.3.3.3). У подібній симуляції наявні всі можливості реального пристрою, а саме:

- Доступ до усіх базових налаштувань і функцій телефону;
- Можливість встановити розроблюваний додаток;
- Доступ до усіх Google-сервісів;
- Доступ до глобальної мережі інтернет;
- Можливість змінювати орієнтацію телефону;
- Можливість увійти у свій Google-акаунт.

Завдяки даній утиліті розробник має можливість змінювати елементи розроблюваного додатку і відслідковувати зроблені зміни у режимі реального часу.

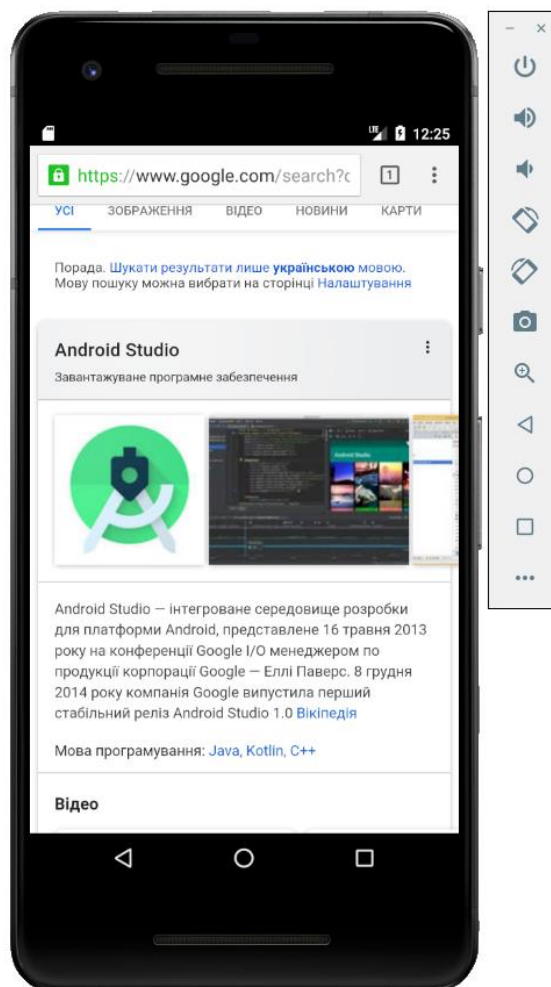


Рис 2.3.3.3 - Демонстрація роботи віртуального пристрою

#### 2.3.4. Помічник налагоджувача Logcat

На думку автора, Logcat є важливим інструментом налагоджувача, бо даний інструмент відображає повідомлення логів, що розсилаються за допомогою різних методів (Рис 2.3.4.1). В Android є спеціальний клас `android.util.Log`, який дає можливість розбивати повідомлення по категоріям в залежності від категорії важливості:

					ІАЛЦ. 045490.004 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підп.	Дата		



- Log.e() – використовується для помилок (error);
- Log.w()- використовується для попередження (warning);
- Log.i()- використовується для інформації (info);
- Log.d()- використовується для налагоджувача (debug);
- Log.v()-використовується для подробиць (verbose);
- Log.wtf()- використовується для позначення виключення, що не повинно було відбутися (What a Terrible Failure)[5].

Для відображення необхідної інформації інструмент Logcat дозволяє створити різні фільтри, пріоритети, відображення повідомлень згенерованих тільки кодом та пошук по самому списку повідомлень.

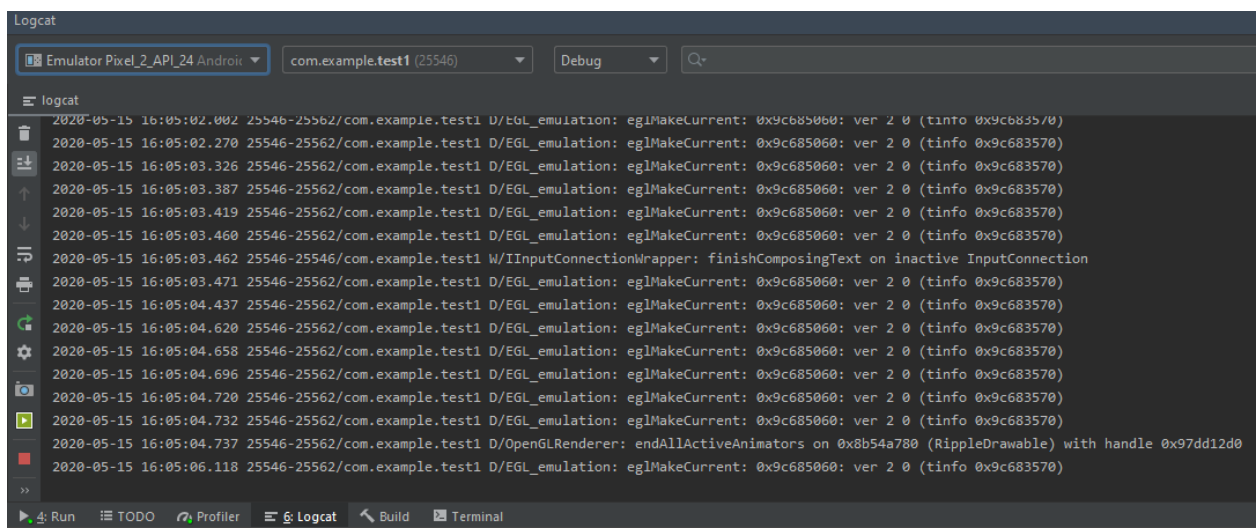


Рис 2.3.4.1- Приклад роботи інструменту Logcat

## 2.4. Особливості розробки програмного інтерфейсу програми

### 2.4.1. Формування програмної частини за допомогою компоненту Activity

Activity – частина додатку для відображення екрану, з яким взаємодіє користувач для виконання певних операцій. Кожній операції присвоюється певна графічна структура для побудови відповідного графічного інтерфейсу. Даний інтерфейс може використовувати як і весь екран пристрою, так і його частину, або розташовуватись поверх других активних вікон. Додаток, як

					ІАЛЦ. 045490.004 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підп.	Дата		

правило, складається з декількох операцій зв'язаних між собою. Одна з операцій є основним робочим столом додатку, який бачить користувач щойно запустивши додаток. Дана головна операція має переходи на інший рівень, тобто, на іншу операцію, яка, в свою чергу, формує свій робочий стіл і має свої функції та графічний інтерфейс. Після запуску нової операції, попередня операція зупиняється, проте зберігається в стеці переходів назад. Стек переходів назад працює по принципу LIFO – процес, що зайшов у стек останнім, вийде першим. Тому після завершення поточної операції, і натиснення кнопки, що повертає назад, поточна операція видаляється зі стеку та знищується, і запускається операція попередня. Для створення операції, необхідно створити підклас класу Activity. Android Studio пропонує розробнику велику кількість базових шаблонів для створення класу Activity (Рис. 2.4.1.1).

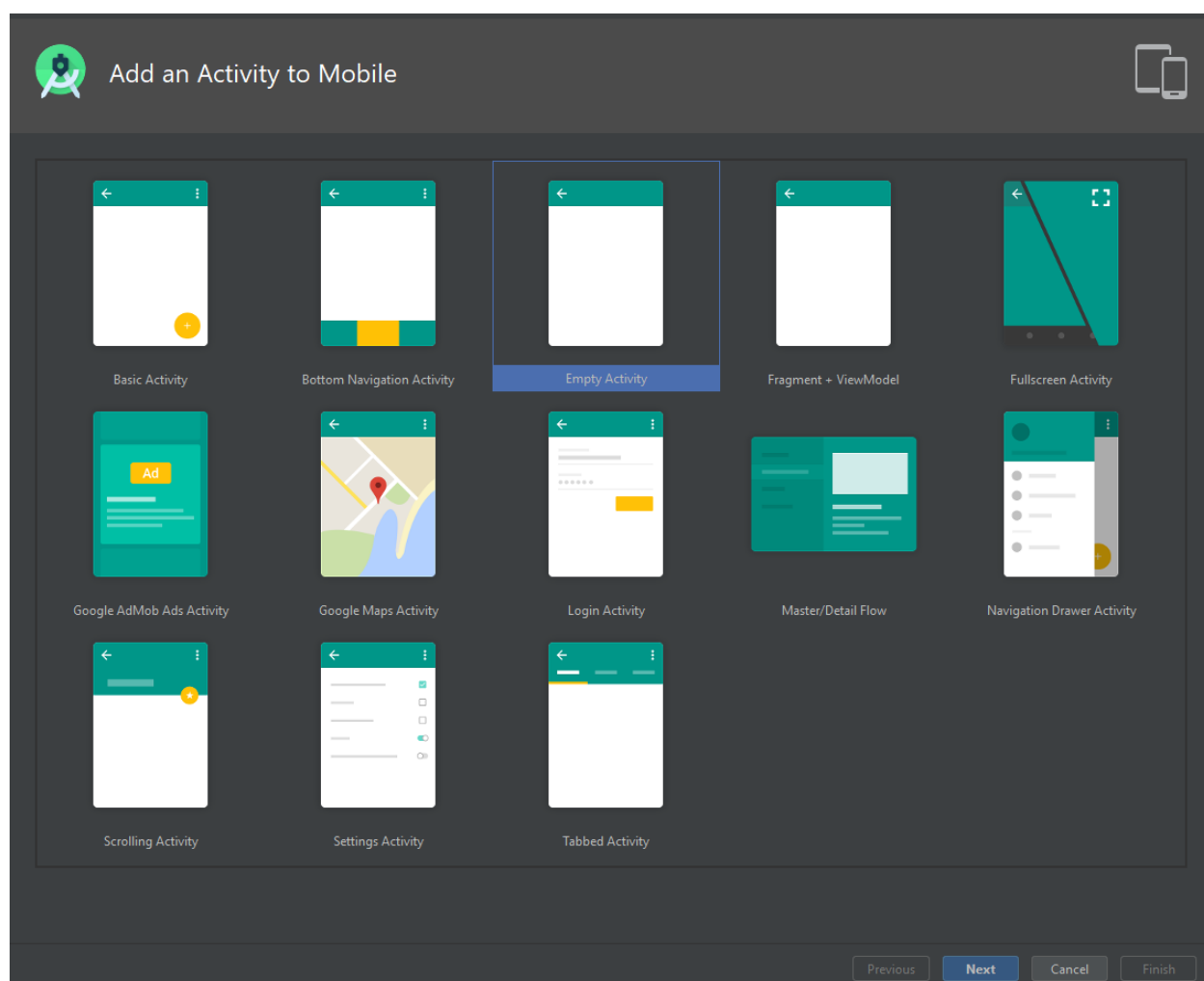


Рис. 2.4.1.1 - Базовий набір шаблонів для створення класу Activity

					ІАЛЦ. 045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		23

В даному підкласі необхідно реалізувати методи зворотного виклику, ці методи система використовує при переході операції з одного стану в інший. Серед усіх існуючих методів зворотного виклику є декілька важливих:

- Метод onCreate() – необхідний метод, оскільки при створенні операції, система викликає саме цей метод. При реалізації необхідно задати ключові компоненти операції. Важливим серед них є метод setContentView(int), який задає макет графічного інтерфейсу операції, та findViewById(int) для використання елементів графічного інтерфейсу у програмному коді та addContentView() – метод, що дозволяє додати будь-який компонент до уже існуючого графічного інтерфейсу;
- Метод onPause() – система викликає цей метод у разі переходу користувача до нової операції. В цьому методі задаються усі зміни, що потребують збереження крім поточного сеансу [6].

Для реалізації переходу між операціями необхідно викликати метод startActivity() і передати в даний метод об'єкт Intent, котрий описує операцію, яку користувач має намір запустити. Для запуску потрібної операції, описується намір котрий явно описує необхідну нам операцію за допомогою імені класу(Рис. 2.4.1.2).

```
Intent intent = new Intent(getApplicationContext(), StudyActivity.class);
startActivity(intent);
```

Рис. 2.4.1.2 - Спосіб описання об'єкту Intent..

Проте може виникнути така ситуація, для виконання якої потрібно передати додаткові дані, які необхідні для коректного функціонування наступної по порядку операції. В цьому випадку розробнику необхідно використати об'єкт Intent для передачі даних в іншу активність(Рис. 2.4.1.3).

Завдяки функції putExtra у розробника є можливість додати певну інформацію у передачу даних до іншої операції. Для реалізації графічного інтерфейсу операції використовується ієрархія елементів – об'єктів, отриманих

					ІАЛЦ. 045490.004 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підп.	Дата		

з класу View. Кожен об'єкт відповідає за певну область у графічному інтерфейсі і може реагувати на дії користувачів. Наприклад, якщо об'єктом є кнопка, то натискання призводить до виконання певної операції.

```
Intent intent2 = new Intent(getApplicationContext(), RepeatWeightActivity.class);  
intent2.putExtra( name: "exerciseId", adapter.getItem(position).getId());  
intent2.putExtra( name: "trainId", trainId);  
startActivity(intent2);
```

Рис. 2.4.1.3 - Спосіб передачі додаткових даних за допомогою об'єкту Intent.

## 2.4.2. Формування основної графічної частини інтерфейсу.

Основна графічна частина додатку формується за допомогою макет. Існує два способи задати макет:

- Оголошення елементів графічного інтерфейсу у XML-файлі. Android Studio пропонує розробнику зручний перелік усіх доступних XML-елементів для класу View і його підкласів.
- Створення елементів у процесі роботи додатку за допомогою певних методів.

Платформа Android наділяє можливістю використовувати будь-який спосіб для оголошення елементів графічного інтерфейсу. Після додавання елемента, за допомогою певних функцій розробник має можливість задати процедуру, яку буде виконувати елемент.

Перевагою оголошення графічного інтерфейсу у XML-файлі є те, що для розробника існує можливість оцінити та змінити розташування певних елементів без запуску компілятора і емулятора( Рис. 2.4.2.1).

Також існує можливість створення адаптацій інтерфейсу під різні платформи, без необхідності змінювати програмним кодом властивості графічного об'єкту. Достатньо створити різні XML-файли, які будуть

відповідати різним розмірам екрану, різним орієнтаціям екрану та різним мовам.

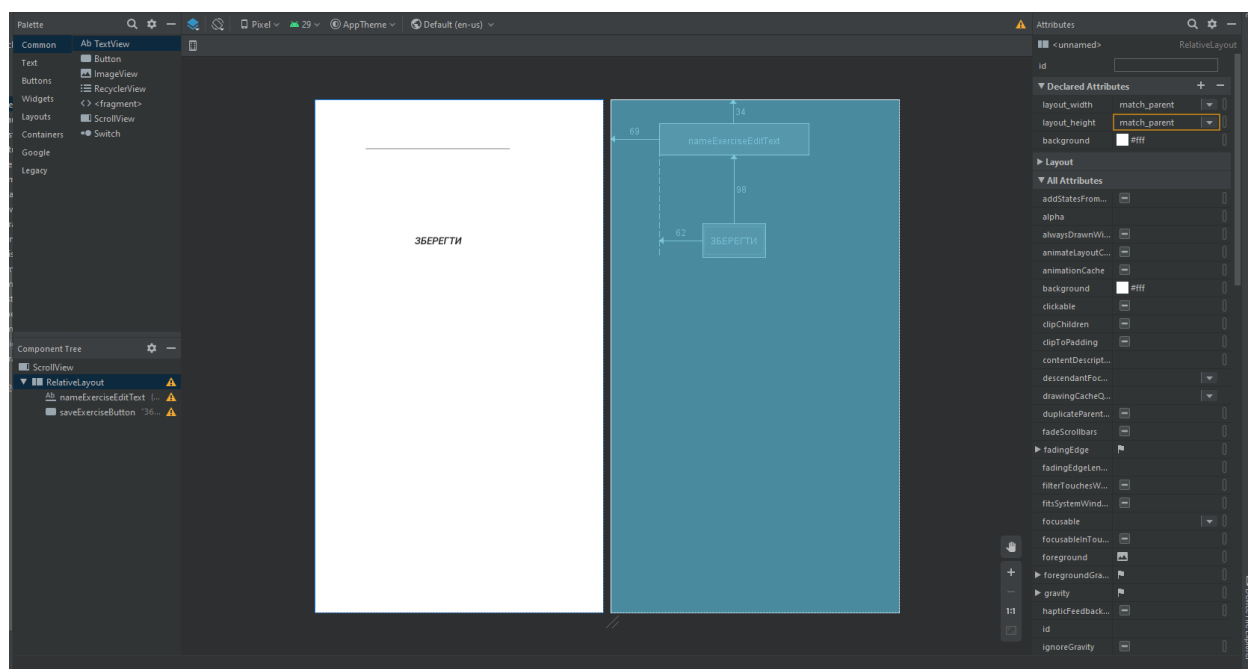


Рис. 2.4.2.1- Відображення елементів графічного інтерфейсу та їх налаштування у XML-файлі.

Під час компіляції додатку кожен XML-файл компілюється в ресурс View. Для того, щоб завантажити макет в код додатку, необхідно викликати метод `setContentView`, і в ньому вказати посилання на макет у наступній формі - `R.layout.activity_add_exercise`. Кожен об'єкт View має ідентифікатор, який існує для позначення унікальності. Згодом цей ідентифікатор можна використати при оголошенні операції, яку виконує певний об'єкт.

Для параметрів ширини та висоти об'єкту можна вказати точні значення, але цей спосіб не завжди є коректним, адже на даний момент у пристроїв різні розміри екрану, і точне оголошення розмірів може негативно вплинути на комфорт користувача. Для цього задаються константи:

- `wrap_content` – розмір активного вікна задається розміром його вмісту;
- `match_parent` – розмір активного вікна задається обмеженнями, які задає його батьківська група.

Активний робочий стіл має прямокутну форму. Розташування активності визначається за допомогою його координат зліва та зверху. Розташування за допомогою даних координат можна отримати шляхом виклику методів `getLeft()` і `getTop()`. Також для більш коректного відображення і позиціонування об'єкту існують методи `getRight()` і `getBottom()` [7].

					ІАЛЦ. 045490.004 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підп.	Дата		

### 3. РЕАЛІЗАЦІЯ ФУНКЦІЇ ЗБЕРЕЖЕННЯ ІНФОРМАЦІЇ ANDROID-ДОДАТКУ ЗА ДОПОМОГОЮ СУБД SQLITE

#### 3.1 Опис системи управління базами даних SQLite

SQLite – це система організації баз даних, яка не використовує парадигму клієнт-сервер, бо є вбудованою. Дана система, на відміну від інших баз даних побудованих на SQL, не має окремого процесу серверу і проводить операції читання та запису даних звертаючись напряду до файлів, що зберігаються на диску. SQLite є найбільш поширеною СУБД у світі, з великої кількості проектів, побудованих на ній.

Серед переваг даної системи найбільшою є її компактність. Об'єм зайнятої пам'яті пристрою може бути меншим за 600 КБ, в залежності від платформи компілятора та розрядності коду. Іноді неправильні налаштування компілятора можуть суттєво збільшити розмір зайнятої пам'яті. Швидкодія даної системи залежить від компілятора та кількості виділеної пам'яті. SQLite є безкоштовною та вільно розповсюджуваною системою, готовою до використання у різних цілях. Дані, якими оперує дана система зберігаються в одному файлі на тому пристрої, де і виконується взаємодія з базою даних.

Проте база даних, створена на одному пристрої, може бути перенесена на інший пристрій. Бази даних побудовані на SQLite можуть коректно функціонувати на пристроях з різною розрядністю та різною архітектурою. Проте основним недоліком цієї бази даних є неможливість зберігання дати у форматі Дата-Місяць-Рік та навпаки [8].

#### 3.2 Реалізація бази даних на Android за допомогою SQLite

Реалізація бази даних на Android за допомогою SQLite використовує абстрактний клас SQLiteOpenHelper. Бібліотека цього класу є стандартною бібліотекою, що міститься в Android Studio. Даний клас надає можливість відкривати, оновлювати, створювати та оперувати базами даних.

					ІАЛЦ. 045490.004 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підп.	Дата		

Для виконання певних операцій з базами даних, SQLiteOpenHelper має декілька методів [9]:

- onCreate – головний метод, викликається при створенні бази даних;
- onUpgrade – даний метод викликається при необхідності оновлення бази даних. Зачасту onUpgrade використовують для додавання таблиць, видалення та певних інших дій, які можуть призвести до оновлення бази даних. При виникненні помилки, всі зміни будуть відмінені;
- onDowngrade – даний метод викликається при необхідності завантажити попередню версію бази даних. Працює ідентично до методу onUpgrade;
- onOpen – даний метод викликається для відкриття бази даних. Цей метод викликається після налаштування зв'язку з базою даних та виконання певних функцій зв'язаних з нею. Наприклад: onCreate або onUpgrade;
- onConfigure – метод існує для налаштування зв'язку з базою даних для підключення функцій запису з упередженням та підтримки зовнішнього ключа. Цей метод викликається перед викликом методів onCreate, onUpgrade, onDowngrade та onOpen;
- close – метод закриває будь-яку відкриту базу даних;
- getDatabaseName – метод повертає назву відкритої бази даних;
- getReadableDatabase - метод існує для відкриття або створення бази даних, відкриває доступ до бази даних у режимі читання, повертає значення бази даних, поки не викликається метод getWritableDatabase або метод close;
- getWritableDatabase - метод існує для відкриття або створення бази даних, відкриває доступ до бази даних у режимі читання та запису;
- setWriteAheadLoggingEnabled – метод існує для підключення або виключення функції упередженого запису. Даний вид запис не може бути використаний для баз даних відкритих у режимі читання або з подібними обмеженнями.



База даних на платформі SQLite використовується автором додатку для читання і запису даних пов'язаних з темою проекту. На думку автора, дана система управління базою даних корисна для всіх Android розробників, через можливість її існування у автономному режимі, без підключення до глобальної мережі інтернет.

					ІАЛЦ. 045490.004 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підп.	Дата		

## 4. ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ANDROID- ДОДАТКУ ДЛЯ КОНТРОЛЮ НАВЧАЛЬНОГО ТА СПОРТИВНОГО ПРОЦЕСУ.

### 4.1 Реалізація головного меню додатку

Головне меню додатку(Рис. 4.1.1) виконане у мінімалістичному дизайні складається з трьох кнопок:

- кнопка «Тренування» – відкриває користувачеві меню тренувань;
- кнопка «Навчання» – відкриває користувачеві доступ до списку навчальних занять створених користувачем;
- кнопка «Календар» – перенаправляє користувача на календар, де у режимі реального часу є можливість слідкувати за розпорядком дня.

Програмна реалізація переходу на наступний робочий стіл відбувається за допомогою методу `setOnClickListener`, але для його використання необхідно зв'язати графічну кнопку та програмну за допомогою методу `findViewById`, який дозволяє використати будь-яку кнопку, що існує у графічному інтерфейсі.

Необхідно створити у XML-файлі елемент, що буде виконувати властивості кнопки. Для даних цілей існує масштабна бібліотека усіх XML-елементів. Після вибору типу кнопки, потрібно присвоїти їй той ідентифікатор, який у подальшому ми хочемо використовувати. Для уникнення плутаниці рекомендується давати кнопкам ідентифікатори зв'язані з їх функціями та призначенням у даній програмі.

Після присвоєння елементу `button` певної кнопки, з'являється можливість підключити обробник натискання через метод `setOnClickListener`. Для зчитування натискань також необхідно задати у XML-файлі властивість кнопки `onClick`. Після цього у Java-класі сформується метод `onClick`, в якому розробник прописує подальші дії, що виконуються після натискання на кнопку.

Для переходу на інший робочий стіл використовуємо об'єкт `intent` та присвоюємо йому певний робочий стіл, на який потрібно направити користувача. Створені таким чином кнопки, з посиланнями на інші робочі столи, утворюють головний робочий стіл програми.

					ІАЛЦ. 045490.004 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підп.	Дата		



Рис. 4.1.1 Головне меню додатку.

## 4.2 Реалізація системи контролю спортивного процесу

### 4.2.1 Реалізація меню вибору тренувань та вправ

Після натискання кнопки Тренування, користувач потрапляє у меню вибору та переходу на подальші функції додатку(Рис 4.2.1.1). Дане меню вибору реалізоване аналогічним чином, з використанням звичайних кнопок, та методу пересилання на наступний робочий стіл.

На робочому столі є такі кнопки меню:

- «Список занять» – кнопка, яка пересилає користувача до списку тренувань;
- «Список вправ» – кнопка, яка пересилає користувача до списку усіх доступних вправ.



Рис. 4.2.1.1 Меню вибору.

Для повернення на попередній робочий стіл використовується метод `onBackPressed`, який дозволяє задати будь-який клас, будь-який робочий стіл до якого повернеться користувач після натискання клавіші або кнопки, що повертає користувача назад. В даному додатку даний метод повертає на попередній робочий стіл з метою відсутності зациклення, яке виникає при багаторазовому натисканні кнопки назад.

#### 4.2.2 Реалізація відображення та редагування списку тренувань

Потрапляючи на даний робочий стіл, користувач бачить список вже існуючих тренувань, функцію для створення нових тренувань та редагування вже існуючих (Рис. 4.2.2.1). Для коректного відображення списку тренувань та вправ автор додатку зв'язує динамічний масив `ArrayList`, який має властивість змінювати свій розмір під час роботи програми, та інтерактивний список

елементів ListView за допомогою адаптера ArrayAdapter. Даний адаптер був створений для роботи з елементами типу ListView, Spinner та подібним. ArrayAdapter робить можливими коректне відображення масиву у графічній частині.

Зв'язується графічна частина з програмною за допомогою метода findViewById. Відповідно для забезпечення взаємодії з кожним елементом списку, використовується метод setOnItemClickListener, який працює ідентично до setOnClickListener, тільки замість одного елемента загалом, оброблює натискання на кожен елемент списку.

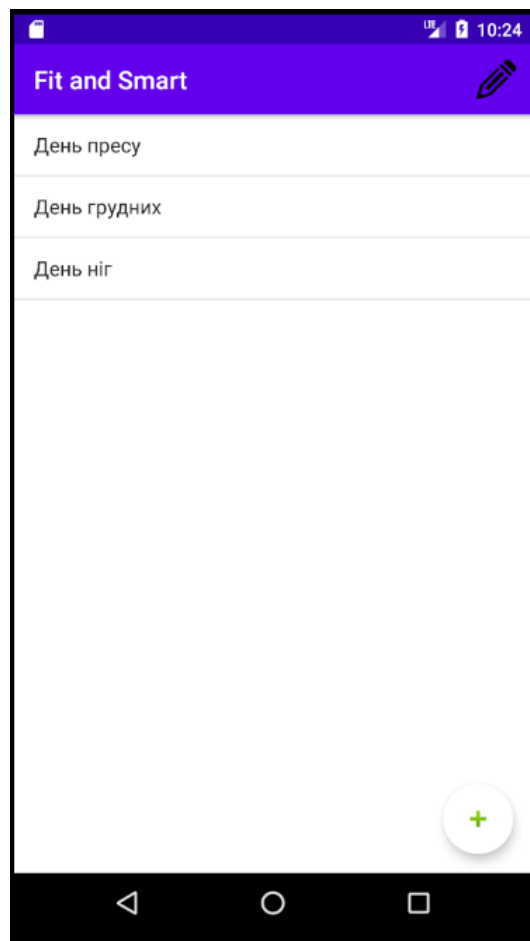


Рис. 4.2.2.1 Список доступних тренувань.

Для перегляду вправ, які входять до складу кожного тренування необхідно натиснути на назву вправи. Після виконання даної дії додаток відкриє список усіх вправ, які є у даному тренуванні (Рис. 4.2.2.2). Даний список працює

					ІАЛЦ. 045490.004 ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підп.	Дата		

аналогічно до списку тренувань , використовуючи ArrayList для збереження даних та елемент ListView для їх відображення .

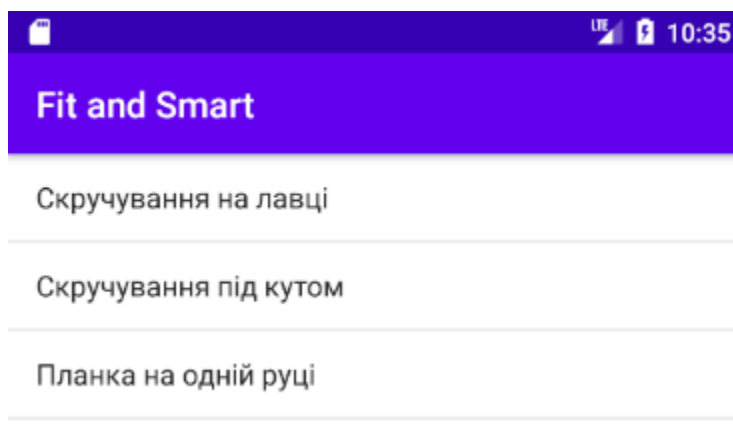


Рис. 4.2.2.2 Відображення списку вправ у тренуванні

Для оголошення елемента, який буде знаходитись у вигляді пункту в меню параметрів необхідно використати та задати метод onCreateOptionsMenu, вказавши у ньому, вибраний розробником, клас MenuItem. Функція обробки натискань пункту меню параметрів реалізована за допомогою методу onOptionsItemSelected, який дозволяє визначити дії додатку після натискання пункту(Рис. 4.2.2.3).



Рис. 4.2.2.3 Меню параметрів додатку

Після натискання на елемент у вигляді олівця, перед користувачем відкриється меню редагування, з допомогою якого можна видалити будь-яке існуюче тренування, за допомогою натискання на червоний хрестик біля назви кожного тренування(Рис. 4.2.2.4). Функція видалення тренування працює за допомогою допоміжного класу TrainHelper, який має методи для взаємодії з базою даних.

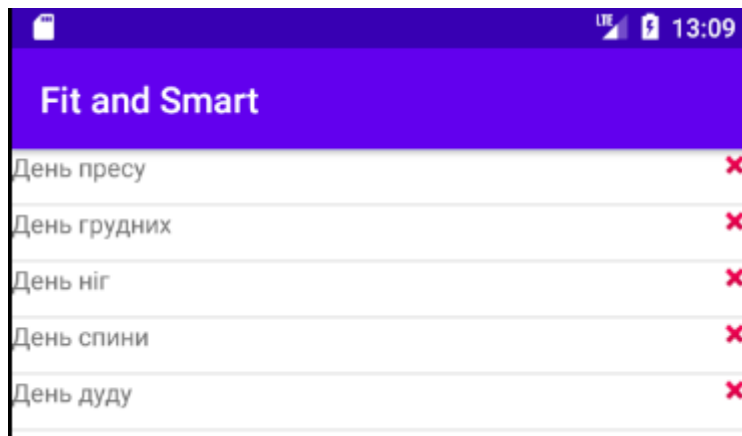


Рис. 4.2.2.4 Функція видалення тренування.

Після натискання, за допомогою функції AlertDialog з'являється маленьке вікно, яке попереджає про користувача про можливість видалення тренування. Якщо користувач натискає кнопку «Ні», то план тренування залишається у базі даних, якщо кнопку «Так», то план тренування видалюється, і користувачеві виводиться повідомлення про успішне видалення тренування. Подібне повідомлення реалізоване за допомогою об'єкту Toast з використанням методу Toast.makeText(), який дозволяє задати для виводу на екран будь-яке повідомлення. Для виводу повідомлення на екран необхідно використати метод show().

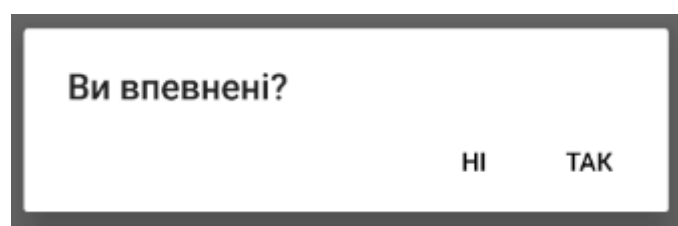


Рис. 4.2.2.5 Отримання згоди користувача на видалення плану тренування.

Після натискання кнопки, схожої по формі на символ «+», запускається функція створення нового тренування. Дана функція реалізована за допомогою функції AlertDialog, яка дозволяє створити невелике вікно, поверх уже існуючого, для запису необхідної інформації (Рис. 4.2.2.6). Дане вікно може мати заголовок, 2-3 кнопки або список із елементів, які можна вибрати. У даному

додатку дане вікно використовується для вибору назви та дати проведення тренувального або навчального заходу.

Для запобігання вибору існуючої назви, автором була розроблена перевірка, яка виключає дану можливість, та виводить користувачеві повідомлення про помилку і прохання назвати тренування інакше. Дана перевірка працює завдяки створеному класу TrainHelper, та методу пошуку вже існуючої назви в базі даних.

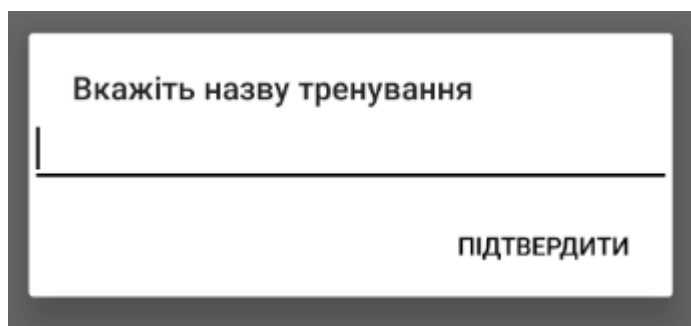


Рис. 4.2.2.6 Вікно для запису назви тренування.

Після введення назви, необхідно натиснути кнопку «Підтвердити», для запису назви тренувань у базу даних. Після цього користувач опиниться на робочому столі вибору вправ для даного тренування.

Даний стіл реалізований за допомогою елементу ArrayAdapter, який транслює ArrayView у графічний елемент ListView(Рис. 4.2.2.7). Для вибору багатьох найменувань, у програмній частині необхідно використати метод setChoiceMode, і встановити значення ListView.CHOICE\_MODE\_MULTIPLE. Таке значення встановить множинний вибір тренувань.

Крім вправ, що створені користувачем, у базі даних додатку вже є деяка кількість стандартних вправ, які знайомі кожному відвідувачу тренажерних залів. Для вибору потрібної вправи необхідно помітити її, шляхом натиснення на квадратик біля назви вправи. Вибравши одну, або декілька вправ, користувач має натиснути кнопку «Додати вправу» для запису інформації про кількість вправ та їх найменування у базу даних.



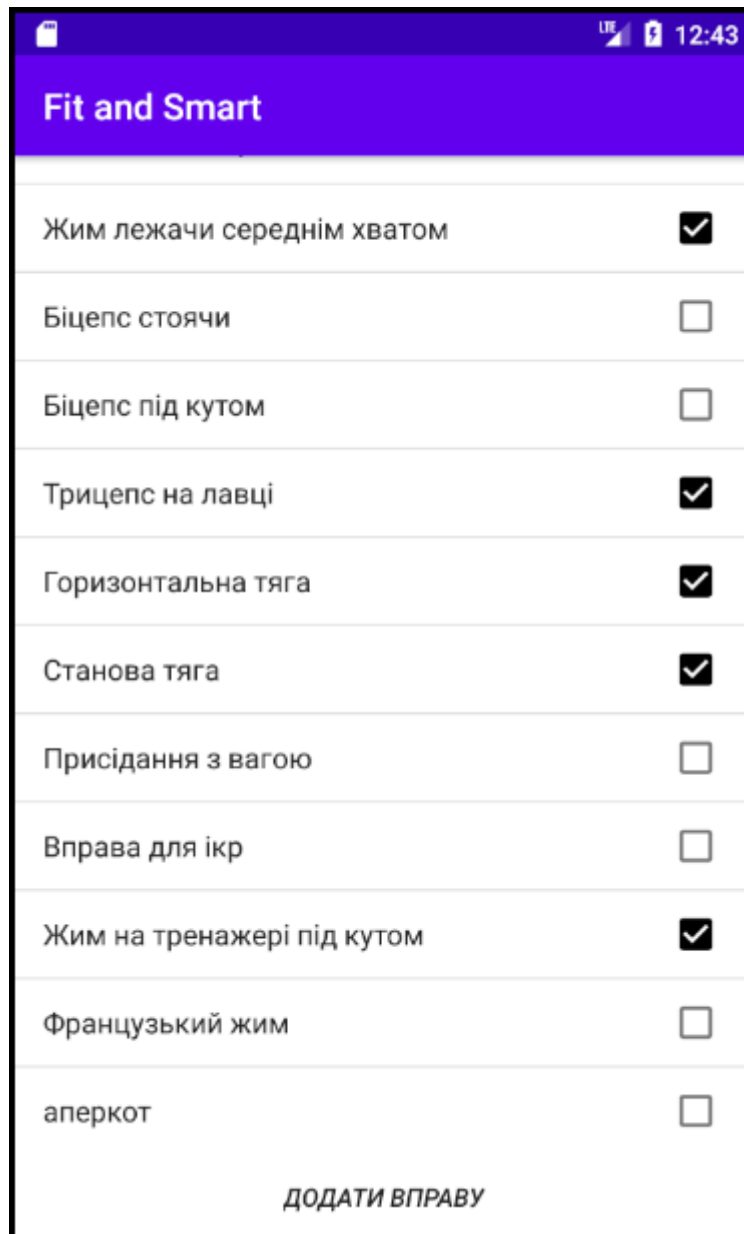


Рис. 4.2.2.7 Список доступних вправ для тренування

Після натискання кнопки «Додати вправу», перед користувачем постане невелике вікно для вибору дати тренування, сформоване функцією AlertDialog (Рис. 4.2.2.8).

Після вибору дня тренування, користувач потрапляє на сформований список вправ, які він вибрав для свого тренування, де натиснувши на назву тренування користувач може вибрати кількість повторень, кількість підходів та робочу вагу кожного підходу. Функція вибору ваги реалізована за допомогою

елементу Spinner. Даний елемент дозволяє відображати лише одне значення серед декількох, на відміну від ListView, який відображає весь масив повністю(Рис. 4.2.2.9).

Рис. 4.2.2.8 Вікно вибору дня тренування.

Проте так, як розроблюваний додаток призначений не тільки для важкоатлетів, тому існує меню вибору константи. Користувач може задати, як кількість підходів, так і кількість кіл у кілометрах або кількість секунд при виконанні вправи на витривалість. Дана функція дозволяє збільшити сферу розповсюдження додатку.

Для видалення зайвих підходів, існує функція для видалення зайвих підходів, яку можна активувати натисканням на червоний хрестик. Після опису необхідної кількості підходів та повторень, потрібно натиснути на символ дискети, яка збереже усі введені дані, і поверне користувача у меню переліку вправ нового тренування для можливості опису підходів у інших вправах(Рис.

4.2.2.10). Проте, якщо вийти з цього вікна шляхом натиснення на клавішу назад, дані не запишуться у базу даних, і тренування можна буде налаштувати заново.

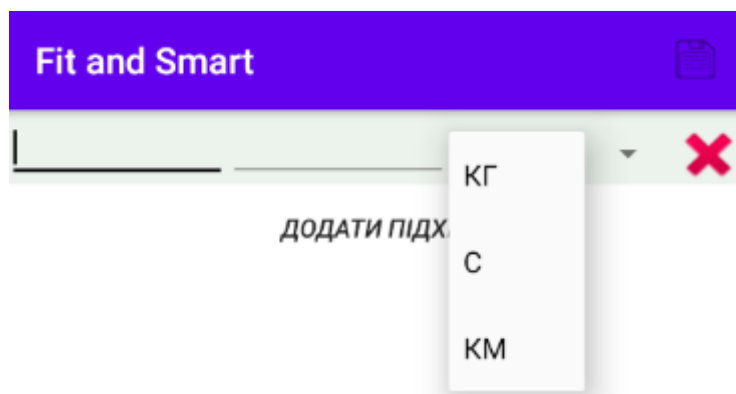


Рис. 4.2.2.9 Можливість вибору константи.

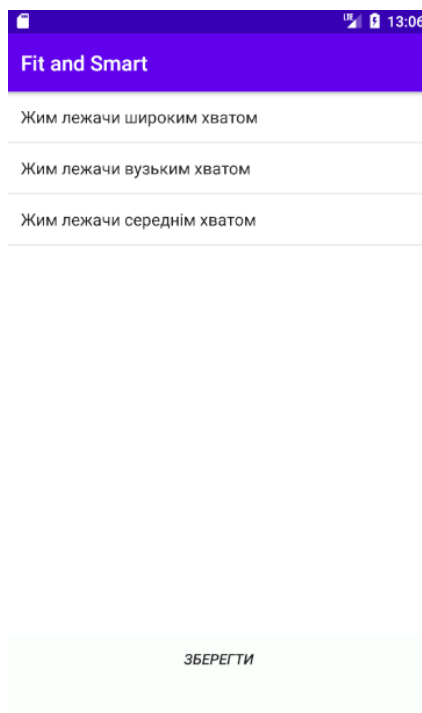


Рис. 4.2.2.10 Меню підтвердження списку тренувань після внесення даних.

### 4.2.3 Реалізація відображення та редагування списку вправ

Потрапляючи на робочий стіл списку тренувань, користувач має можливість створити нове тренування, або видалити вже існуюче(Рис. 4.2.3.1).

					ІАЛЦ. 045490.004 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підп.	Дата		

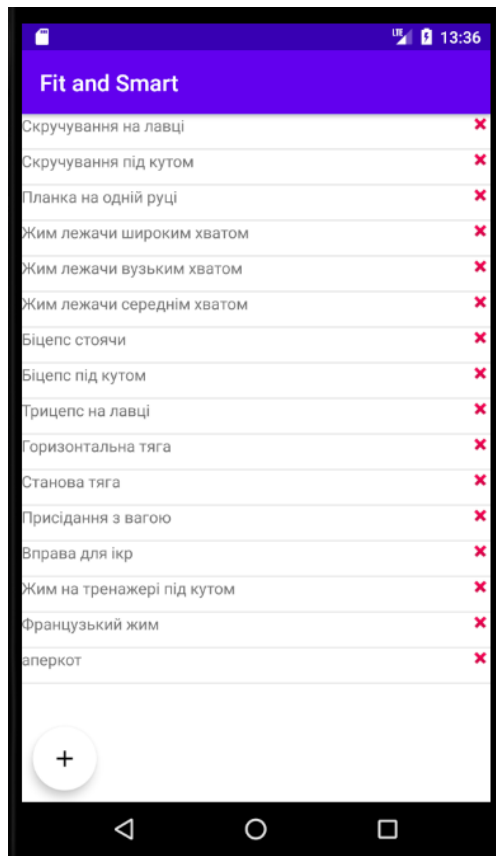


Рис. 4.2.3.1 Інтерфейс списку тренувань

Натиснувши на червоний хрестик навпроти назви вправи, користувач має змогу видалити її. Після цього перед користувачем з'явиться вікно з підтвердженням видалення, і при підтвердженні, обрана вправа видалиться з бази даних.

Для створення нової вправи необхідно натиснути «+». Після цього користувач опиниться на новому робочому столі, і вказавши назву нової вправи зберегти її у базі даних (Рис. 4.2.3.2). При внесенні вже існуючої вправи користувач буде попереджений про недоцільність своїх дій. Дане повідомлення реалізоване за допомогою об'єкту Toast.

Рис. 4.2.3.2 Інтерфейс створення нової вправи.

### 4.3 Реалізація системи контролю навчального процесу

Після вибору у головному меню кнопки «Навчання», користувач потрапляє до робочого столу з відображенням списку предметів (Рис. 4.3.1). Принцип роботи даного столу схожий з принципом роботи робочого столу списку тренувань. За допомогою зв'язки елемента ListView та ArrayAdapter виконується відображення списку предметів.

Так, як назва кожного предмету унікальна, то у базі даних відсутні будь-які записи. Користувач сам може заповнити даний список згідно з списком предметів, або речей, на яких він хоче акцентувати свою увагу.

Користувач може записати у базу даних новий предмет. Для цього користувач має натиснути кнопку «+» та ввести назву предмета у форму. Після цього натиснути на кнопку «Вибрати день тижня», та вибрати день тижня для нового предмету.

Для видалення необхідно натиснути на елемент у меню інструментів, який відповідає за редагування та перейти на робочий стіл видалення найменування (Рис. 4.3.2). Принцип видалення предмету є подібним до видалення тренування, проте модуль Study використовує свій клас для взаємодії з базою даних StudyHelper, який виконує всі дії, пов'язані з базою даних предметів.

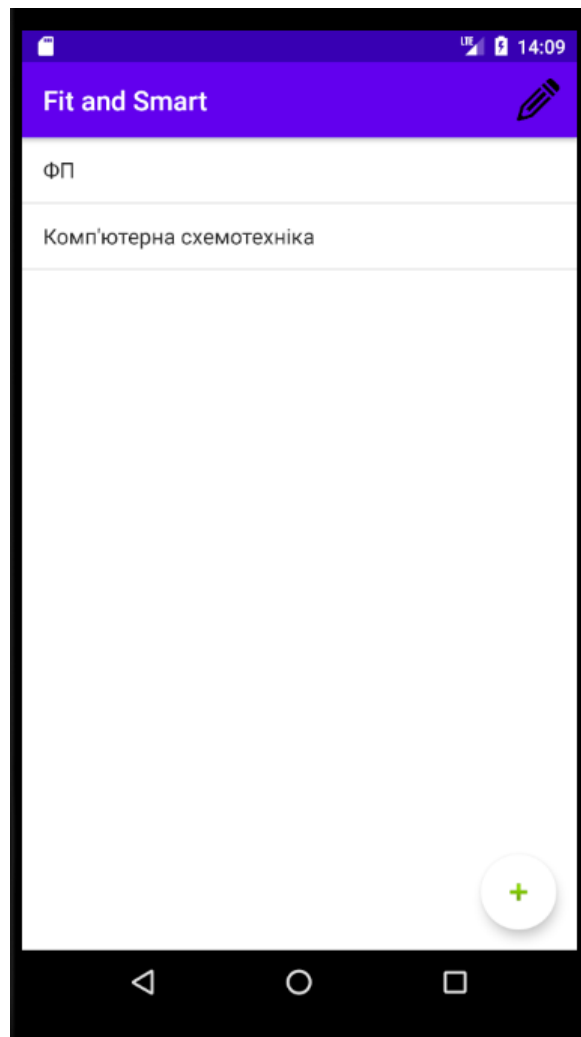


Рис. 4.3.1 Інтерфейс відображення списку предметів

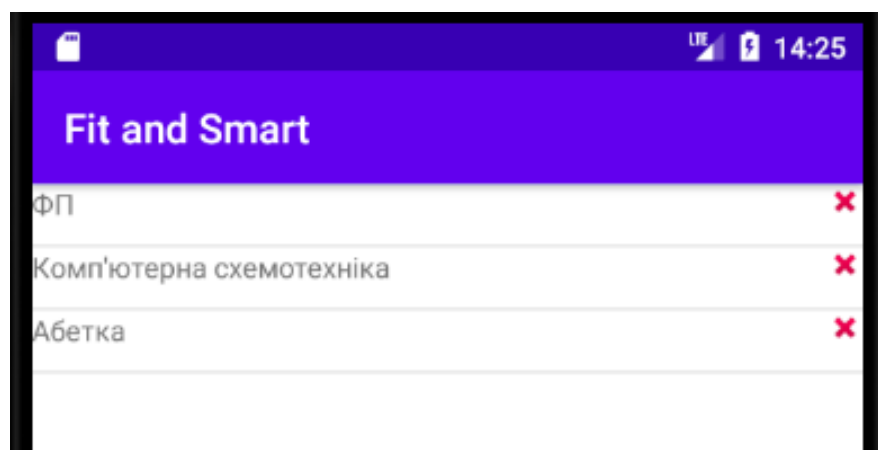


Рис. 4.3.2 Відображення функції видалення предмету

#### 4.4 Реалізація системи моніторингу за тренуваннями та навчальною діяльністю

Після вибору у головному меню кнопки «Календар», користувач потрапляє на робочий стіл з календарем (Рис. 4.4.1).

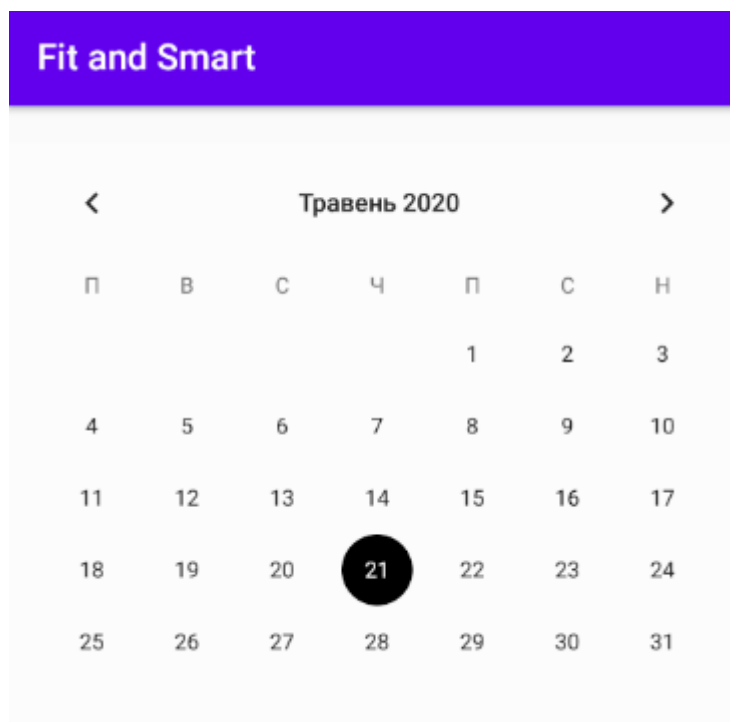


Рис. 4.4.1 Інтерфейс для вибору дати

Реалізація цього модулю є можливою завдяки великій бібліотеці елементів Android. Цей модуль є одним із базових, і потребує мінімального налаштування. Для календарю, як і для звичайної кнопки необхідно підключити властивість `onClick`, яка дозволить оброблювати натискання на елементи календарю і за допомогою методу `setOnDateChangeListener` реагувати на вибрану дату. Після вибору дати запускається клас, який відображає список справ на день у режимі реального часу.

Проте через те, що SQLite не підтримує зберігання дати виникає проблема, як саме синхронізувати дату календарю і день тижня, на який вже записано тренування. Вирішення цієї проблеми полягає у збиранні лише дня тижня та конвертації його у порядковий номер, який потім використовується як

Зм.	Арк.	№ докум.	Підп.	Дата

ІАЛЦ. 045490.004 ПЗ

Арк.  
44

ідентифікатор дня тижня. Згодом цей ідентифікатор допомагає знайти усі записи бази даних, які призначаються саме обраному дню.

#### 4.5 Реалізація та конфігурація бази даних тренувань за допомогою класу TrainHelper

Для коректного функціонування бази даних у додатку, автором був створений клас TrainHelper, який виконує функції створення, редагування та читання бази даних SQLite. Правильне зберігання даних потребує наявності чотирьох таблиць:

- RepeatTable – таблиця повторень, основними її даними є кількість повторень, вага, кілометраж або час;
- ApproachTable – таблиця, яка зберігає інформацію про кількість підходів, і дані попередньої таблиці;
- ExerciseTable – таблиця, яка з кількості підходів формує вправу, зберігає її назву, та порядковий номер кожної вправи;
- TrainTable – таблиця тренувань, яка містить назву тренування, порядковий номер тренування, день тижня для тренування.

Використовуючи ці 4 таблиці TrainHelper за допомогою методів створює базу даних в якій зберігається вся потрібна користувачеві інформація. На початку роботи з базою даних задається її назва, а також оголошуються усі ключові елементи, за допомогою яких і буде здійснюватися взаємодія з базою даних.

Після створення бази даних, за допомогою методів задаються різні функції, необхідні нам: функції для додавання елемента, для пошуку по порядковому номеру тренування, для пошуку по значенню дня тижня, перевірка на вже існуючий елемент, для підрахунку підходів. Дана перевірка виконується для знаходження вже існуючого тренування та вправи, завдяки цій перевірці працює функція відображення вже існуючих даних.



Саме в класі TrainHelper автор проєкту використовує помічник налагоджувача Logcat, який дозволяє відслідковувати помилки різних типів та виводити повідомлення про знайдену помилку у Log повідомлень. Саме ця утиліта є дуже корисною на етапі налагоджування і тестування, та дозволяє швидко і просто вказати на помилку, що може виникнути у процесі компіляції коду.

Для управління базою даних, що містить інформацію про предмети та їх дні розташування використовується клас StudyHelper, який функціонує ідентично до класу TrainHelper, проте для оперування базою даних використовує тільки одну таблицю Study.

					ІАЛЦ. 045490.004 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підп.	Дата		

## ВИСНОВКИ

На сьогоднішній день кожна восьма людина на планеті Земля має пристрій на платформі операційної системи Android. Більшість цих людей веде активний спосіб життя, і серед такої більшості є ті, хто потребує простого у дизайні та використанні додатку, який супроводжуватиме їх у активному житті.

На ринку цифрових послуг існує багато різних додатків подібного типу. Проаналізувавши такі додатки, автор проєкту поділив існуючі додатки на декілька типів, порівнюючи їх між собою та загальним шаблоном додатку даного типу. Перший тип додатків перевантажений рекламою і зайвими функціями, інші мають весь необхідний функціонал, але платні. Існують і такі додатки, мета яких – збирати та зберігати дані про користувача з метою нав'язування йому рекламних оголошень, що відповідають його інтересам. На період сьогоднішнього, серед усього цього цифрового мотлоху, простому користувачеві дуже важко віднайти той самий простий і надійний безкоштовний додаток, не перевантажений зайвими функціями та рекламними оголошеннями. До того ж неможливо віднайти додаток, який вирішує дві задачі: ведення спортивного та учбового щоденника та його відображення у режимі реального часу.

Метою даного дипломного проєкту було розроблення додатку, який має задовольнити усі вимоги, зазначені вище. Після аналізу можливостей операційних систем, та їх можливостей була вибрана операційна система Android, яка є більш лояльною до стороннього розробника та надає власну колекцію бібліотек і середовище розробки Android Studio. Дане середовище ідеально підходить для початківця, бо має інструменти та підказки, що допомагають початківцю швидко і правильно розібратися з усіма можливостями програми. Ще однією перевагою даного середовища є наявність емулятор віртуальних пристроїв, який дозволяє відслідковувати відображення додатку на різних пристроях.

Створений додаток, дає можливість користувачам занотовувати план свої тренувань та навчання у пам'ять телефону. Також даний додаток має мінімалістичний дизайн, що дуже популярно у сьогоденні, та простий, інтуїтивно зрозумілий інтерфейс. Основною цільовою аудиторією цього додатку є школярі старших класів, студенти і просто люди з активною життєвою позицією, які не бояться кидати собі виклик і сміливо дивляться у майбутнє.

					ІАЛЦ. 045490.004 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підп.	Дата		

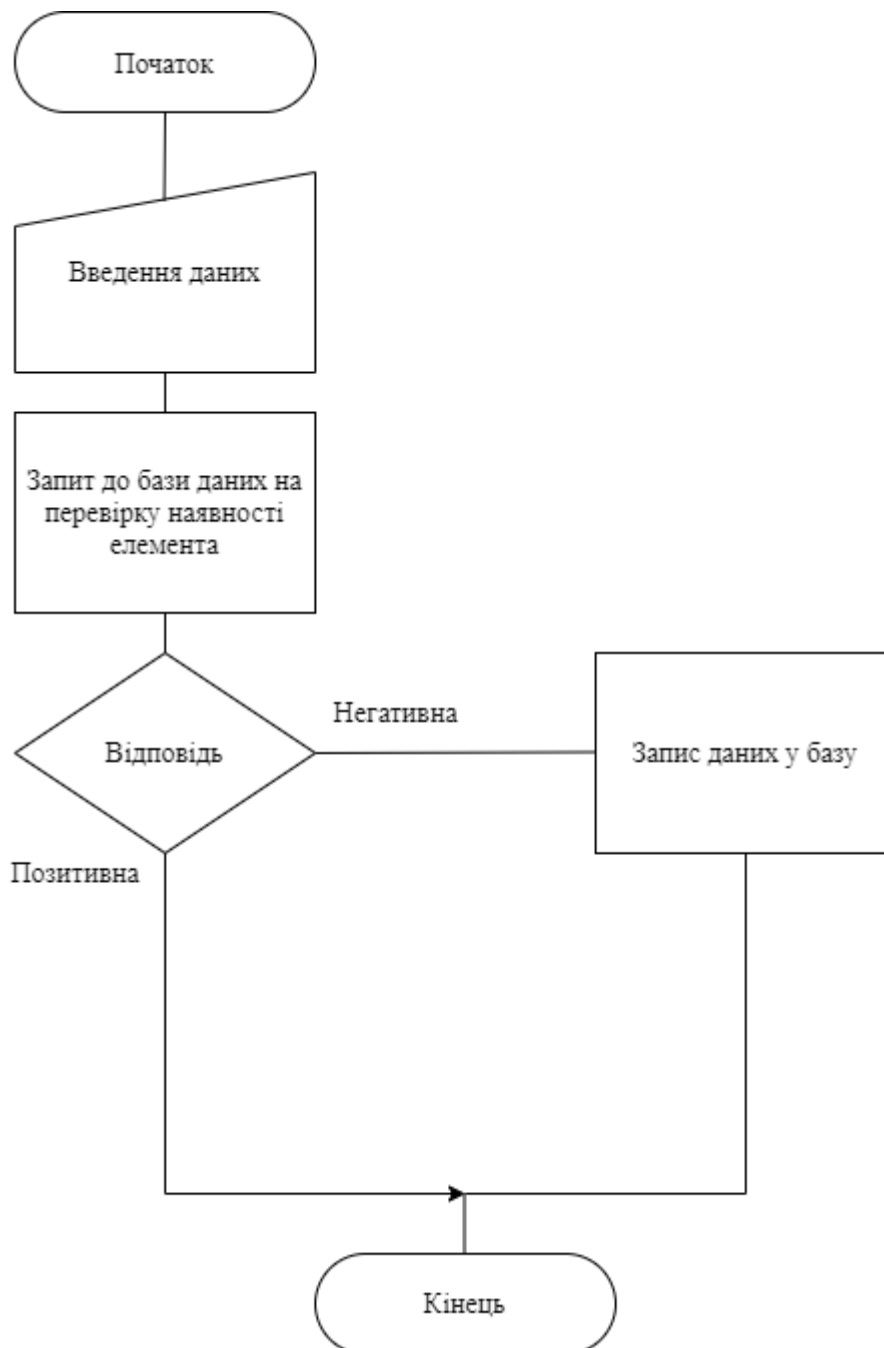
## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Історія компанії Android [Електронний ресурс]. - Режим доступу: <https://www.androidauthority.com/history-android-os-name-789433/>(дата звернення: 12.04.2020 ).
2. Про Android Studio [Електронний ресурс]. - Режим доступу: <https://developer.android.com/studio/intro> (дата звернення: 15.05.2020 ).
3. Принцип роботи Gradle. [Електронний ресурс]. - Режим доступу: [https://docs.gradle.org/current/userguide/what\\_is\\_gradle.html#what\\_is\\_gradle](https://docs.gradle.org/current/userguide/what_is_gradle.html#what_is_gradle) (дата звернення 14.05.2020).
4. Пакет Android SDK [Електронний ресурс]. - Режим доступу: <https://developer.android.com/guide/components/fundamentals?hl=ru> (дата звернення: 14.05.2020 ).
5. Logcat та його функціонал [Електронний ресурс]. - Режим доступу: <https://developer.android.com/reference/android/util/Log> (дата звернення: 14.05.2020 ).
6. Клас Activity та його особливості [Електронний ресурс]. - Режим доступу: <https://developer.android.com/guide/components/activities?hl=ru> (дата звернення: 24.04.2020 ).
7. Базові графічні функції [Електронний ресурс]. - Режим доступу: <https://developer.android.com/guide/topics/ui/declaring-layout> (дата звернення: 24.04.2020 ).
8. Характеристики та призначення SQLite [Електронний ресурс]. - Режим доступу: <https://www.sqlite.org/about.html>(дата звернення: 29.04.2020 ).
9. Використання SQLiteOpenHelper [Електронний ресурс]. - Режим доступу: <https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html> (дата звернення 29.04.2020).
10. Гріффітс Д., Гріффітс Д., Програмування на Android. 2015. 734 с.
11. Herve J. Franceschi. Android App Development 2015 682 с.
12. raywenderlich Tutorial Team. Saving Data on Android. 2019. 290 с

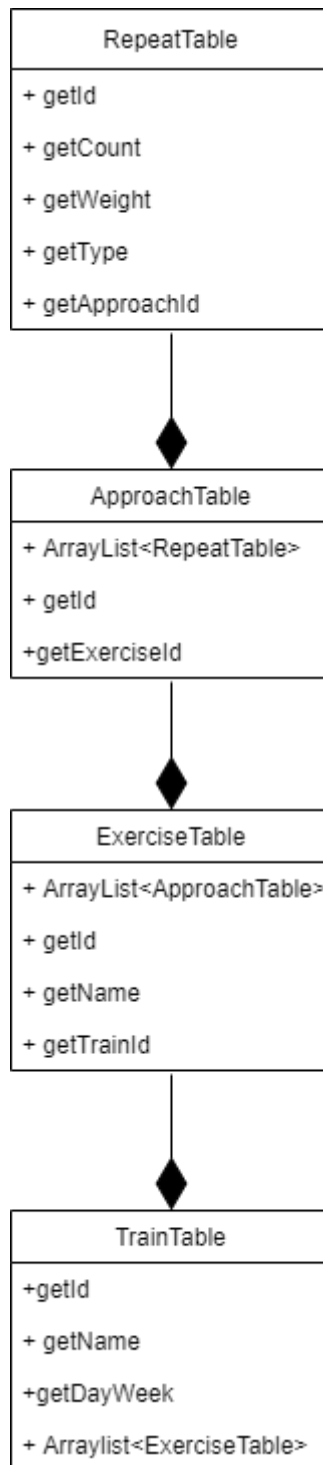
					ІАЛЦ. 045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		49

13. Murphy Mark. The Busy Coder's Guide to Android Development. 2019. 468 c.
14. Murphy Mark. Exploring Android. 2019. 542 c.
15. Windmill Eric. Flutter in Action. 2020. 368 c.
16. Sunny Kumar Aditya., Vikash Kumar Karn., Android SQLite Essentials. 2014. 110 c.
17. Belen Cruz Zapata. Android Studio Essentials. 2015. 126 c.

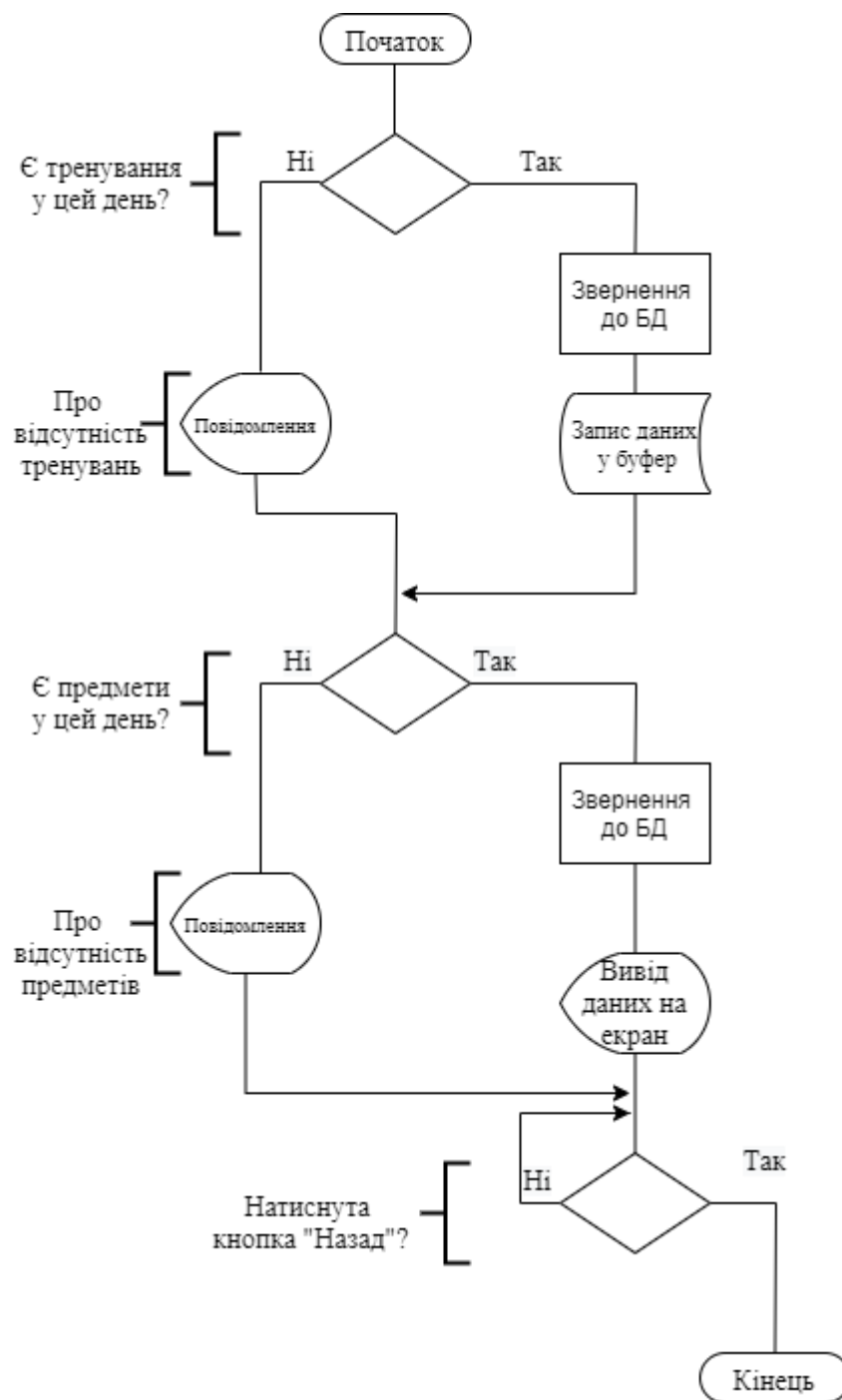
					ІАЛЦ. 045490.004 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підп.	Дата		



					ІАЛЦ.045490.005 Д1						
Зм.	Арк	№ докум	Підпис	Дата							
Розроб.		Філатов О.О.			Запис нового найменування у базу даних. Схема алгоритму			Літ.		Арк.	Аркушів
Перев.		Наливайчук М.В.								1	
								КПІ ім. Ігоря Сікорського, ФПМ КВ-63			
Н. контр.		Клятченко Я.М.									
Затв.		Романкевич. В.О.									

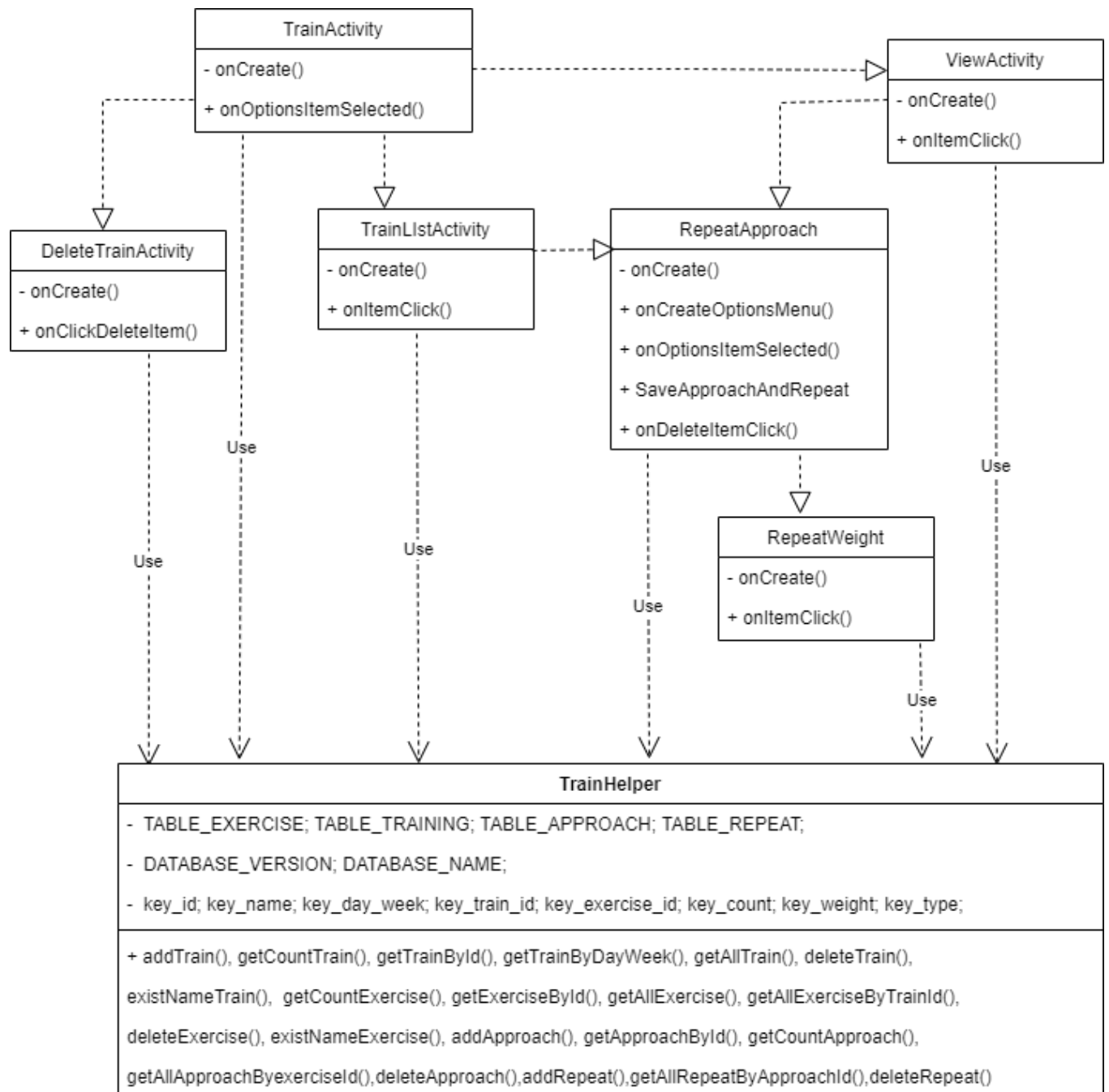


					ІАЛЦ.045490.006 Д2			
Зм.	Арк	№ докум	Підпис	Дата	Зв'язок таблиць, що формують базу даних. Схема структурна			
Розроб.		Філатов О.О.						
Перев.		Наливайчук М.В.						
Н. контр.		Клятченко Я.М.						
Затв.		Романкевич В.О.						
					Літ.		Арк.	Аркушів
							1	
					КПІ ім. Ігоря Сікорського, ФПМ КВ-63			



					ІАЛЦ.045490.007 ДЗ			
Зм.	Арк	№ докум	Підпис	Дата	Вивід поточної події. Схема алгоритму			
Розроб.		Філатов О.О.						
Перев.		Наливайчук М.В.						
Н. контр.		Клятченко Я.М.						
Затв.		Романкевич В.О.						
					Літ.		Арк.	Аркушів
							1	
					КПІ ім. Ігоря Сікорського, ФПМ КВ-63			





					ІАЛЦ.045490.008 Д4							
Зм.	Арк	№ докум	Підпис	Дата								
Розроб.		Філатов О.О.				Літ.			Арк.		Аркушів	
Перев.		Наливайчук М.В.							1			
						КПІ ім. Ігоря Сікорського, ФПМ КВ-63						
Н. контр.		Клятченко Я.М.										
Затв.		Романкевич. В.О.										
					Взаємозв'язки модулю Train. Схема структурна							